# 特別研究報告書

題目

# 格子電極構造の3次元配列イオントラップ

Novel 3D-Arrayed Ion Trap with Electrodes of Lattice Structure

---


指導教員

木村 正廣　教授

報告者

1095304

小野田 有吾


平成19年2月19日


高知工科大学大学院工学研究科基盤工学専攻

電子・光システム工学コース

# 目 次

**第 1 章　序論** **1**

**第 2 章　イオントラップの原理** **2**
2.1 Paul Trap の原理 . . . . . 2
2.2 四重極型 (リニア) イオントラップ . . . . . 5
2.3 安定化領域ダイアグラム . . . . . 7
2.4 Mathieu 方程式 . . . . . 9

**第 3 章　イオントラップの電極構造** **10**
3.1 格子電極構造 . . . . . 10
3.2 帯電液滴生成部 . . . . . 12

**第 4 章　SIMION7.0 によるシミュレーション** **13**
4.1 電極の作製 . . . . . 13
4.2 格子構造をもつ電極のポテンシャル . . . . . 16
4.3 シミュレーション結果 . . . . . 17
4.3.1　シミュレーションによる安定化領域 . . . . . 20

**第 5 章　格子電極構造をもつ**
**イオントラップの製作** **22**

**第 6 章　実験** **26**
6.1 実験装置の全体図 . . . . . 26
6.2 帯電液滴のトラップ . . . . . 27
6.2.1　試作機 1 でのトラップ結果 . . . . . 27
6.2.2　試作機 2 でのトラップ結果 . . . . . 29
6.3 試作機 2 に対する安定化領域ダイアグラム . . . . . 30
6.4 試作機 2 拡張時のトラップ結果 . . . . . 32
6.4.1　トラップした帯電液滴による光散乱 . . . . . 35

**第 7 章　結論** **41**

**付 録 A Mathieu 方程式に関するプログラム** **42**

**付 録 B SIMION で使用したプログラム** **44**
B.1 交流電圧設定プログラム . . . . . 44
B.2 ポテンシャルデータ取り出しプログラム . . . . . 50

**付 録 C Mie 散乱プログラム** **55**

**付 録 D ２段階励起オプトガルバノ分光** **59**

**謝辞** **62**

**参考文献** **63**

# 第1章　序論

イオンは電荷をもっているため電場や磁場から力を受けることを利用し，電磁場を用いてイオンを空間に閉じ込める技術および装置をイオントラップという．イオントラップは２つの種類に分類される．一つはrfイオントラップ(又はポールトラップ)で，交流電場と静電場を用いる手法である．もう一つはペニングトラップで，静電場と静磁場を用いる手法である．残念ながら静電場のみではEarnshowの定理によってイオンをトラップすることができない[1]．

2002年ノーベル化学賞を受賞した田中耕一氏の研究は質量分析器を用いてたんぱく質の解析を行ったものであるが，イオントラップも精密な質量分析器として活用されている．イオンをトラップした後，レーザー冷却をすることで，GPSなど多くの応用が期待できる非常に高精度な原子時計を実現しようとする研究がさかんである[2,3]．

1994年のShorの因数分解アルゴリズムを契機に量子コンピュータの研究が盛んとなり，イオントラップおよびレーザー冷却の技術は，その基本素子である量子ビットを生成するために応用され始めている[4]．イオントラップを使った量子ビットの操作はCirac-Zoller法[5]に基づいており，イオンを１次元に配列させる手法が主流となっている[7,8]．筆者はイオンを３次元的に配列させることができれば，飛躍的な展開が望めるのではないかと考えた．そこで本研究では，イオンを３次元的に配列できるイオントラップの新しい電極構造を提案し，プロトタイプのものを作製してその特性を調べた[*1)]．

なお，本実験で扱った液滴の質量は原子イオンのそれに比べてはるかに大きいが，イオントラップ中での粒子の振る舞いは粒子の質量，電荷，交流電圧と周波数，トラップ空間で表されるパラメータで決まるため，本実験の結果を原子イオンのトラップにスケーリングすることができる．これにより本研究で提案する格子電極構造をマイクロスケールのイオントラップにも適用することができる．

[*1)] 本大学は学位論文を基本的にオンライン化している．この論文も後輩達に読まれることを考え若干，冗長になっている部分もあるがご了承願いたい．

# 第2章　イオントラップの原理

## 2.1　Paul Trapの原理

Paul Trapは図2.1に示すように3枚の回転双曲面からなる電極であり上下の電極をエンドキャップ電極, 中央の電極をリング電極と呼んでいる．このトラップはW.Paulが最初に考案したもので，彼はイオントラップの研究により，1989年のノーベル物理学賞を受賞した[6].

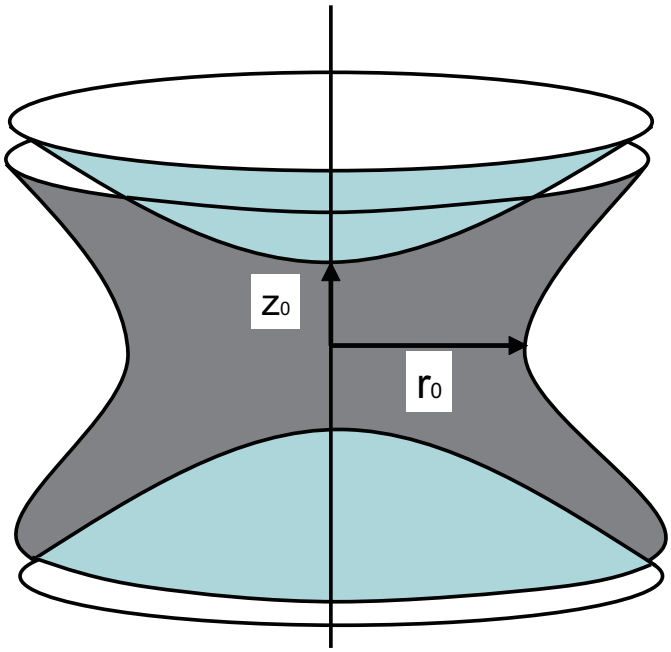



図 2.1: Paul Trap の電極構造.

3次元中の任意のポテンシャルは一般的に下記のように記すことができる．

$$\phi_{x,y,z} = A(\lambda x^2 + \sigma y^2 + \gamma z^2) + C \tag{2.1}$$

上式はラプラス方程式 (2.2), (2.3) を満たさないといけない.

$$\nabla^2 \phi_{x,y,z} = 0 \tag{2.2}$$

$$\nabla^2\phi_{x,y,z} = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial z^2} = 0 \tag{2.3}$$

一方，式 (2.1) より

$$\frac{\partial\phi}{\partial x} = \frac{\partial}{\partial x}(A\lambda x^2) = 2A\lambda x \tag{2.4}$$

y, z に関しても同様にして

$$\nabla^2\phi_{x,y,z} = A(2\lambda + 2\sigma + 2\gamma) = 0 \tag{2.5}$$

$A \neq 0$ なので

$$\lambda + \sigma + \gamma = 0 \tag{2.6}$$

を得る．式 (2.6) を満たす $\lambda$, $\sigma$, $\gamma$ のペアとしては

$$\lambda = -\sigma = 1, \gamma = 0 \tag{2.7}$$

及び,

$$\lambda = \sigma = 1, \gamma = -2 \tag{2.8}$$

がある．式 (2.7) は四重極 (リニア) 型に該当する. これについては 2.2 節で解説する．式 (2.8) が図 2.1 中の任意ポテンシャル式 (2.1) を満たすペアである．よって図 2.1 中の任意ポテンシャルは,

$$\phi_{x,y,z} = A(x^2 + y^2 - 2z^2) + C \tag{2.9}$$

と書くことができる．図 2.1 は z 軸に対して回転対称なので, 式 (2.9) を円柱座標系に直すと，すなわち $x = r\cos\theta$, $y = r\sin\theta$, $z = z$ と置くと式 (2.9) は

$$\phi_{r,z} = A(r^2 - 2z^2) + C \tag{2.10}$$

次に定数 A を決めないといけない．リング電極とエンドギャップ電極のポテンシャルをそれぞれ $\phi_{ring}$ と $\phi_{endcaps}$ としてその差分を $\phi_0$ とおくと

$$\phi_0 = \phi_{ring} - \phi_{endcaps} \tag{2.11}$$

まずリング電極が作るポテンシャルは $z = 0$ で $r = \pm r_0$ なので式 2.10 に代入すると

$$\phi_{ring} = A(r_0^2) + C \tag{2.12}$$

同様に，エンドギャップ電極が作るポテンシャルは $r = 0$ で $z = \pm z_0$ なので

$$\phi_{endcaps} = A(-2z_0^2) + C \tag{2.13}$$

式 (2.12) と式 (2.13) の境界条件を得られるので，式 (2.11) に代入すると

$$\phi_0 = A(r_0^2 + 2z_0^2) \tag{2.14}$$

よって定数 A は

$$A = \frac{\phi_0}{(r_0^2 + 2z_0^2)} \tag{2.15}$$

次に定数であるが，一般的にエンドギャップ電極はグランドに接続しておくので境界条件として次式が得られる．

$$\phi_{0,z_0} = \phi_{endcaps} = \frac{\phi_0(0 - 2z^2)}{(r_0^2 + 2z_0^2)} + C = 0 \tag{2.16}$$

よって C は

$$\therefore C = \frac{2\phi_0 z_0^2}{r_0^2 + 2z_0^2} \tag{2.17}$$

式 (2.15) および式 (2.17) を, 式 (2.10) に代入して

$$\phi_{r,z} = \frac{\phi_0(r^2 - 2z^2)}{(r_0^2 + 2z_0^2)} + \frac{2\phi_0 z_0^2}{r_0^2 + 2z_0^2} \tag{2.18}$$

次に $\phi_0$ を

$$\phi_0 = (U + Vcos\Omega t) \tag{2.19}$$

と置いて現実的な形にする．次にイオンの運動を考えていく．例えば z 方向の運動を考えると

$$\begin{aligned} F_z = -e\left(\frac{d\phi}{dz}\right)_r &= e\frac{4\phi_0 z}{r_0^2 + 2z_0^2} \\ &= m\left(\frac{d^2z}{dt^2}\right) \end{aligned} \tag{2.20}$$

よって式 (2.20) に式 (2.21) を代入して

$$m\left(\frac{d^2z}{dt^2}\right) = \frac{4(U + Vcos\Omega t)z}{r_0^2 + 2z_0^2} \tag{2.21}$$

カッコの中を展開して

$$\frac{d^2z}{dt^2} = \left(\frac{4eU}{m(r_0^2 + 2z_0^2)} + \frac{4eVcos\Omega t}{m(r_0^2 + 2z_0^2)}\right)z \tag{2.22}$$

式 (2.22) は，マシュー (Mathieu) 方程式と呼ばれる形をしており

$$\frac{d^2u}{d\xi^2} = (a_u - 2q_u cos\xi)u = 0 \tag{2.23}$$

と書ける．ただし，

$$a_z = -2a_r = -\frac{16eU}{m(r_0^2 + 2z_0^2)\Omega^2} \tag{2.24}$$

と

$$q_z = -2q_r = \frac{8eV}{m(r_0^2 + 2z_0^2)\Omega^2} \tag{2.25}$$

である．式 (2.23) が発散しない解をもつためには，$a_z$ と $q_z$ が決まった領域になければならない．

## 2.2 四重極型 (リニア) イオントラップ

四重極型のイオントラップは図 2.2 に示すとおり 4 本の電極棒から成っており，対角のものが対を成している．

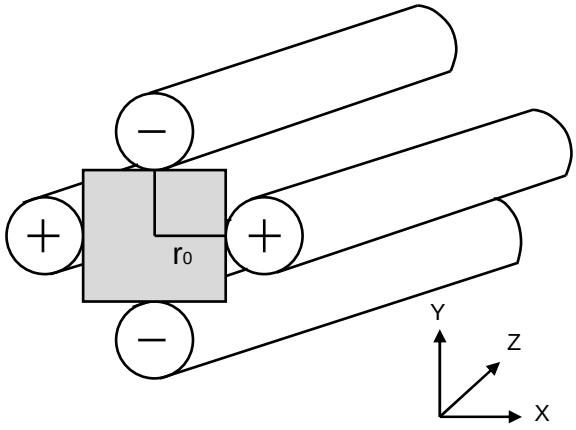



図 2.2: 四重極 (リニアトラップ) 型イオントラップの電極構造.

四重極型は式 (2.7) の場合に相当するので，式 (2.9) に対応するものとして

$$\phi_{x,y,z} = A(x^2 - y^2) + C \tag{2.26}$$

式 (2.11) に対応するものとして

$$\phi_0 = \phi_{x\,pair} - \phi_{y\,pair} \tag{2.27}$$

Paul Trap の時と同様に順次計算していくと

$$\begin{aligned} \phi_{x\,pair} &= A(r_0^2) + C \\ \phi_{y\,pair} &= A(-r_0^2) + C \end{aligned} \tag{2.28}$$

式 (2.27) に代入して

$$\begin{aligned} \phi_0 &= 2A(r_0^2) \\ \therefore A &= \frac{\phi_0}{2r_0^2} \end{aligned} \tag{2.29}$$

そして

$$\phi_{0,0} = C \tag{2.30}$$

であるが，特別なことをしなければ$C = 0$とおける．したがって式 (2.26) は

$$\phi_{x,y} = \frac{\phi_0}{2r_0^2}(x^2 - y^2) \tag{2.31}$$

となる．
次にイオンの運動を知りたいのでまず$y = 0$における x 方向の運動をまず考える．

$$\phi_{x,0} = \frac{\phi_0 x^2}{2r_0^2} \tag{2.32}$$

ポテンシャルが決まれば電場も自ずと決まるので

$$\left(\frac{d\phi}{dx}\right)_y = \frac{\phi_0 x}{r_0^2} \tag{2.33}$$

よって (x,0) においてイオンは以下の力を受ける．

$$F_x = -e\left(\frac{d\phi}{dx}\right)_y = -e\frac{\phi_0 x}{r_0^2} \tag{2.34}$$

力が決まれば運動方程式を立てられるので

$$m\left(\frac{d^2x}{dt^2}\right) = -e\frac{\phi_0 x}{r_0^2} \tag{2.35}$$

現実的には

$$\phi_0 = 2(U + Vcos\Omega t) \tag{2.36}$$

とおいて

$$\frac{d^2x}{dt^2} = -\left(\frac{2eU}{mr_0^2} + \frac{2eVcos\Omega t}{mr_0^2}\right)x \tag{2.37}$$

そして$\xi = \frac{\Omega t}{2}$とおいて変数変換をしてマシュー (Mathieu) 方程式 (2.23) に対応させると

$$\begin{aligned} a_x &= \frac{8eU}{mr_0^2\Omega^2} \\ q_x &= \frac{-4eV}{mr_0^2\Omega^2} \end{aligned} \tag{2.38}$$

が得られる．

## 2.3 安定化領域ダイアグラム

式 (2.24), (2.25) に対応した安定化領域ダイアグラムと呼ばれているものを以下の図 2.3 に示す．図 2.3 において太線内の $a_z$, $q_z$ は式 (2.24)，(2.25) に対して安定な解を持つ．すなわちイオンは安定にトラップされることを意味する．

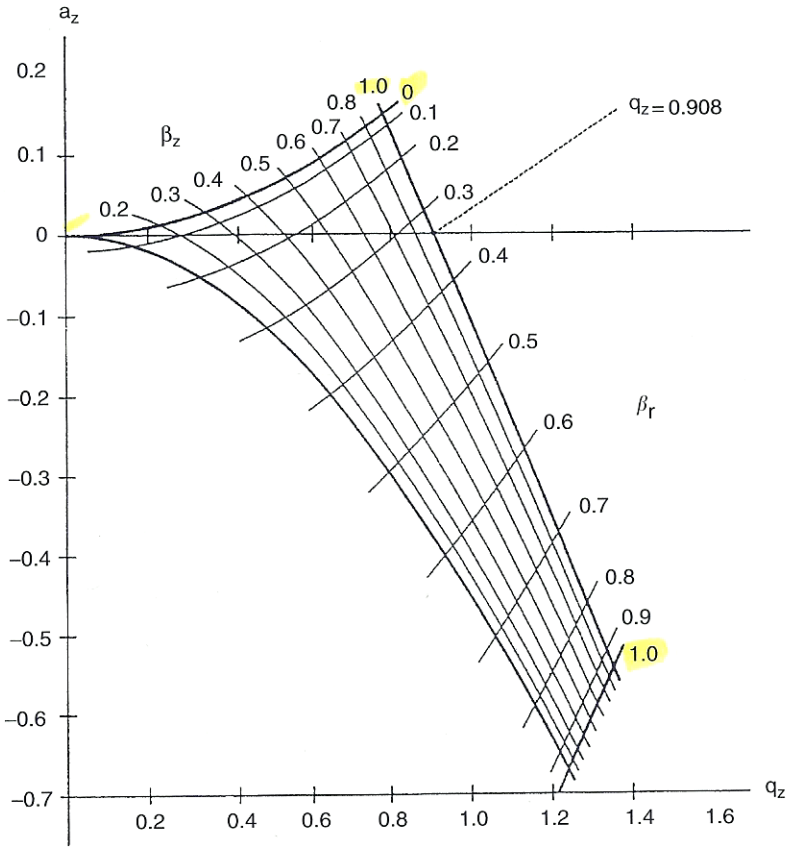



図 2.3: 双曲面型イオントラップの安定化領域ダイアグラム．
参考文献 [1] より引用．

同様にリニアトラップの場合にも式 (2.38) に対応した安定化領域 (斜線部) が存在するので以下の図 2.4 に示す．$x(q$ パラメータ) 軸に対して対照的なのが特徴である．

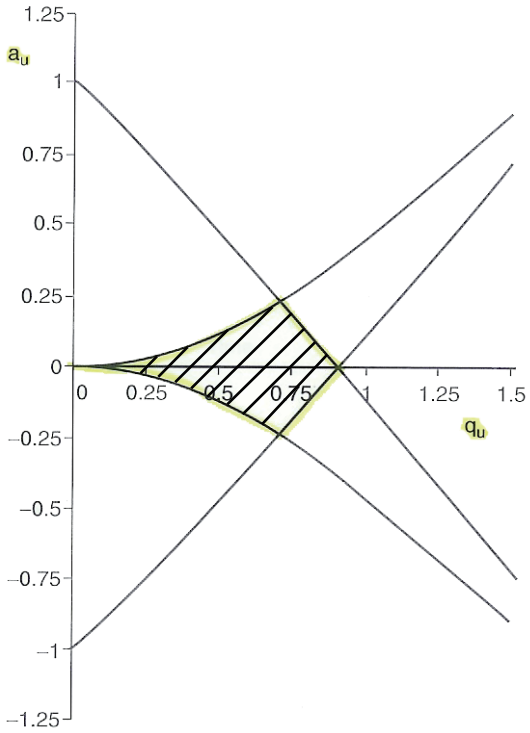



図 2.4: 四重極 (リニアトラップ) 型イオントラップの安定化領域ダイアグラム．参考文献 [1] より引用．

## 2.4 Mathieu方程式

マシュー (Mathieu) 方程式 (2.23) の $a$ パラメータ，$q$ パラメータに実際に値を入れて Mathematica で解いた図 2.5 を以下に示す.

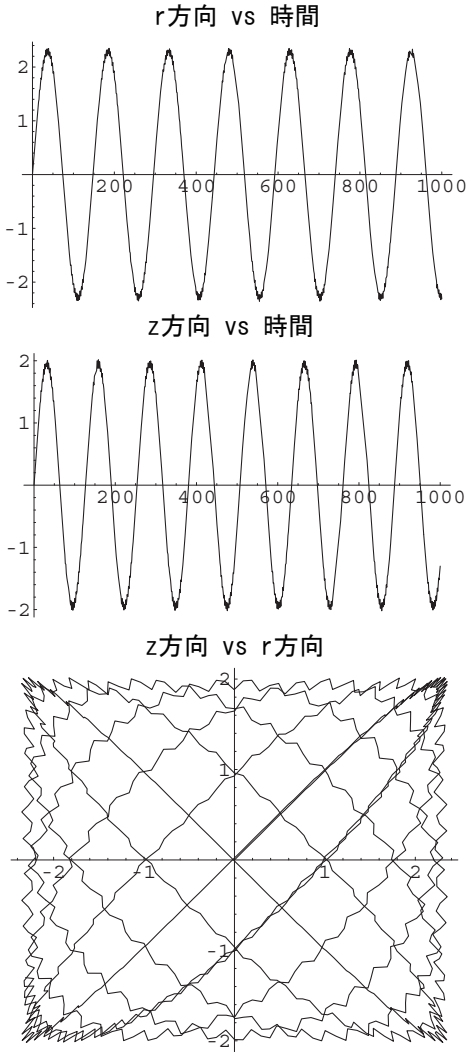



図 2.5: マシュー方程式を解いて得たイオンの軌道.
$a_r = 0, q_r = 0.06, a_z = 0, q_z = 0.07$ とした.

大きな周期が永年運動と呼ばれるものである．小さなギザギザは交流電場をかけていることからくるマイクロモーションと呼ばれる運動である．また，この Mathematica のプログラムを付録 A に記した.

# 第3章　イオントラップの電極構造

## 3.1　格子電極構造

今回，筆者が提案する格子電極構造をもつイオントラップ装置の全体図を図 3.1 に示す.

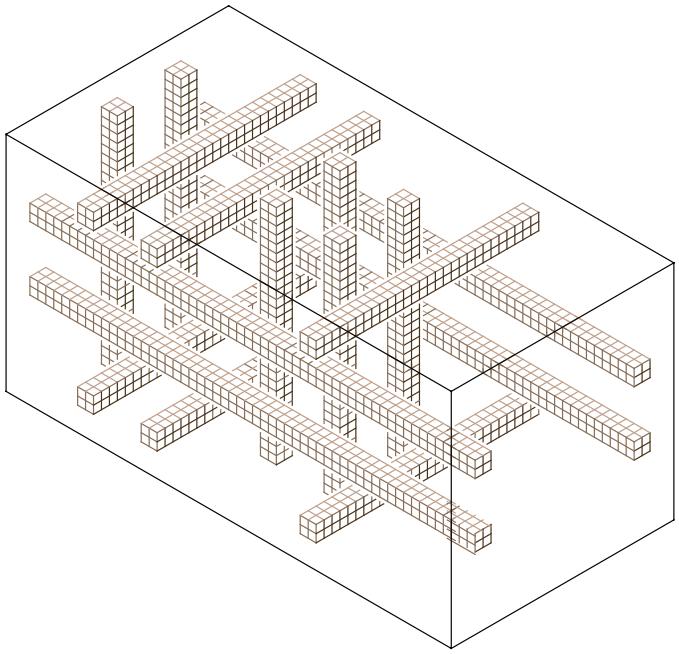


図 3.1: 筆者が提案する格子電極構造をもつイオントラップ装置の全体図.

この電極構造の特徴として，通常の回転層局面型電極構造に比べて空隙が多いためイオンの操作や観測が容易に行えること，トラップしたイオンを 3 次元的に自由に動かすことができ, 隣のセルへの移動が可能であることなどが挙げられる．以下にイオン光学系シミュレーションソフト SIMION7.0(4 章参照) で作成した各セルの電極構造を示す．図 3.1 の構造は図 3.2, 3.3 の構造を組み合わせて作られる．このように図 3.2, 3.3 のセルを追加していくだけでトラップ領域を 3 次元的に拡張できる.

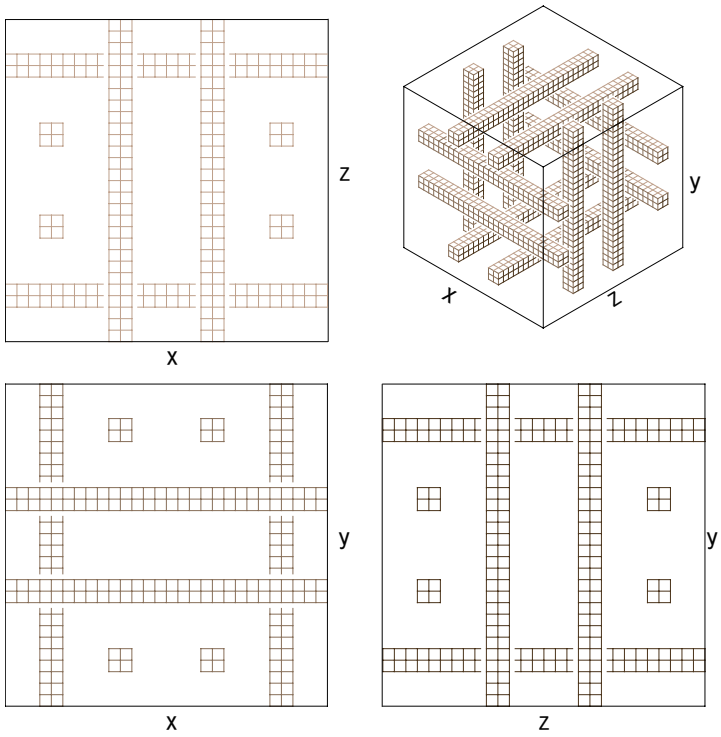



図 3.2: 格子電極構造 (セル) の一部 (その１).

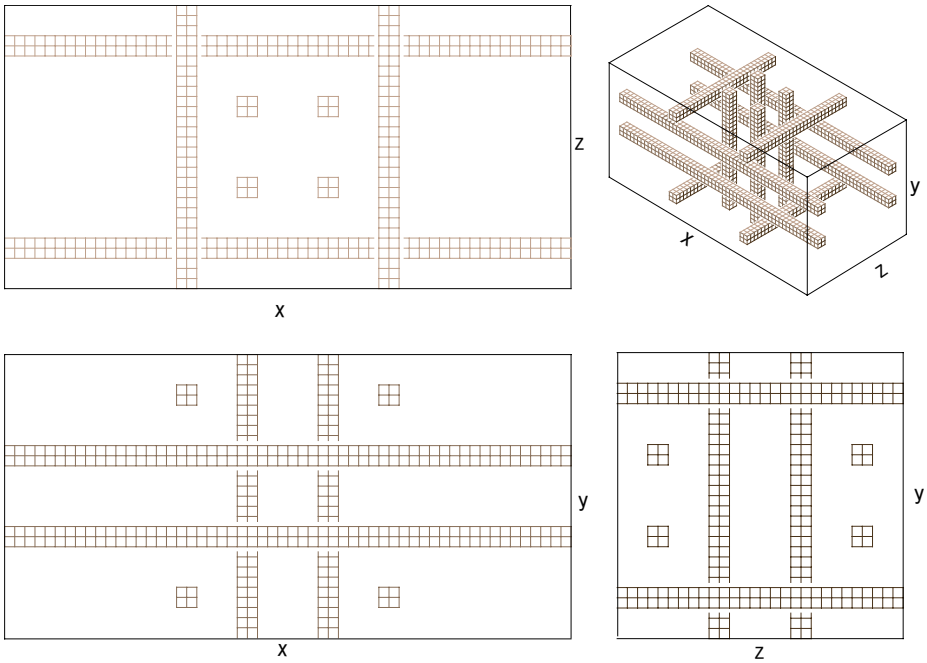



図 3.3: 格子電極構造 (セル) の一部 (その２).

## 3.2 帯電液滴生成部

イオン (帯電液滴) を生成するための電極構造を図 3.4 に示す．これは以前, 本研究室で平行平板型のイオントラップを研究していた時に使用していたものである [8–10]．液体 (本研究では蒸気圧の低いグリセリンを使用) を入れた注射針の先端に高電圧をかけると針の先の液体は帯電する．この帯電した液体は電場による力でが引きちぎられ，電荷を持った液滴すなわちイオンが飛び出ることになる[*1)]．

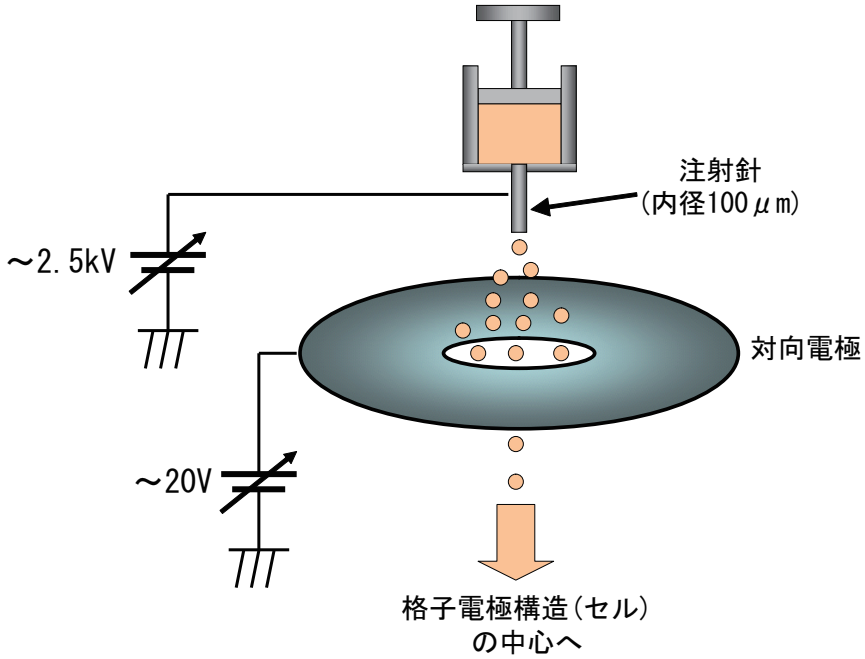



図 3.4: 帯電液滴生成部．注射針にはグリセリンが入っている．高電圧をかけて針の先端でしずくとなった液体を引きちぎることで，プラスに帯電したグリセリン液滴を得る．

[*1)] 英語で Electro-Spray Ionaization(ESI) という．実験時一番苦労したのは注射針と対向電極の距離である．適した $m/q$ の帯電液滴を作製できなければいつまで経ってもトラップできない．近すぎるとバチバチと音を立て放電する．約 0.5mm～1mm 程度が適切かと考えている．条件:温度, 湿度 etc によって適切な距離は変わると思う．ちなみに湿度 50 %の時にもトラップできた．

# 第4章　SIMION7.0によるシミュレーション

## 4.1　電極の作製

SIMION7.0[*1)]とはイオン光学系のシミュレーションソフトである．他にもITSIM, ISISなどがある．SIMIONは, 最も汎用性があり設定できるパラメータ等も多いのが特徴である．図4.1がSIMIONを起動した時のトップ画面である．

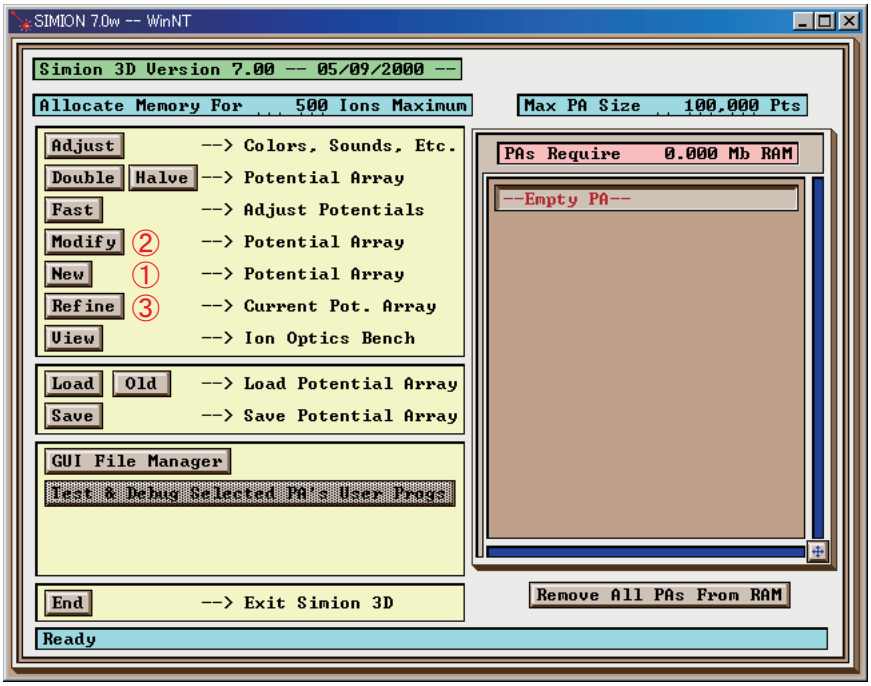



図 4.1: SIMION7.0を立ち上げた時のトップ画面.

新規にファイルを作成する場合には図4.1の　で示したNewをクリックして，作成するグリッド領域を決める．次に　で示したModifyをクリックすると図4.2の画面が現われるので，適宜適切な平面において電極を作製していく．

[*1)] URL:http://www.simion.com/

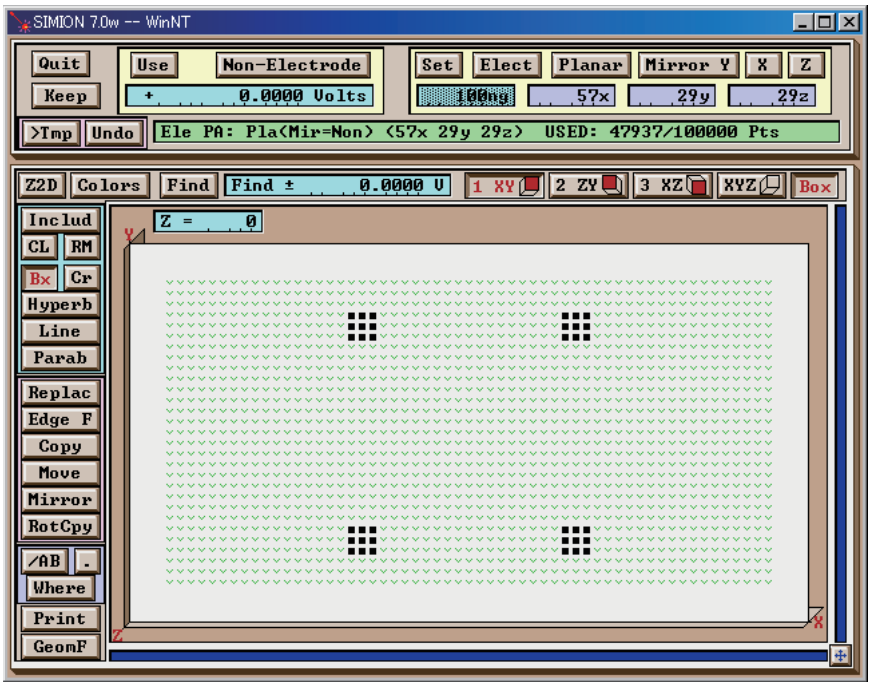



図 4.2: SIMION7.0 で電極をモデリングしていく画面.

このようにして作製していって最終的に完成した直方体型電極構造の全体図が図 3.2 である．電極を作製し終わったら Keep を押して保存する．保存する時には先に保存する先のフォルダを作っておくとよい．***.pa#という名前をつけて保存することに注意したい．図 4.1 のトップ画面で Refine をクリックすると***.pa0 を始めとした，***.pa1〜***.pa 電極数 までのファイルが作られる．

### 画像ファイルの書き出し

SIMION は作製した電極構造をファイルとして出力することができる．出力する時の画面キャプチャが以下の図 4.3 である．電極を作製したファイルと同じフォルダに画像が出力される．

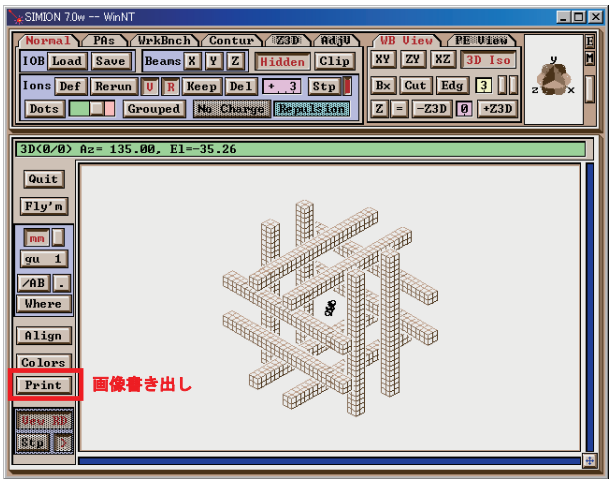



図 4.3: 作製した電極構造をファイルへ出力する画面.

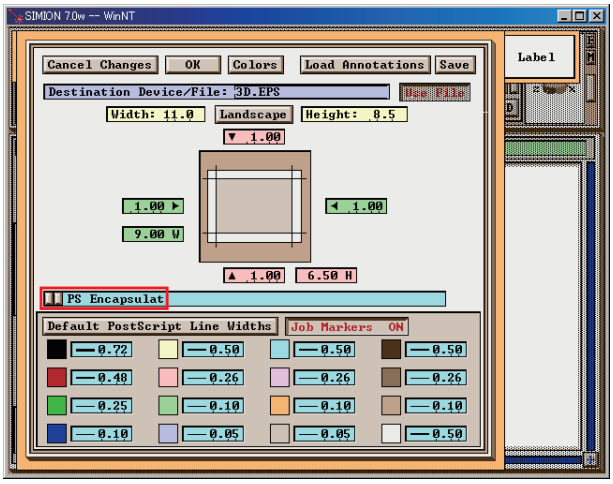



図 4.4: 赤枠で囲んだ部分で出力するファイル形式を選択する．eps ファイルとしても出力できる．

## 4.2　格子構造をもつ電極のポテンシャル

図 3.2 に対応する格子電極について，SIMION7.0 でポテンシャルデータを取り出してグラフ表示させたものが図 4.5 である．3 枚のそれぞれは中心を通る異なった平面上のポテンシャルを見ている．SIMION7.0 から電圧データを取り出すためのプログラムを付録 C にまとめた．

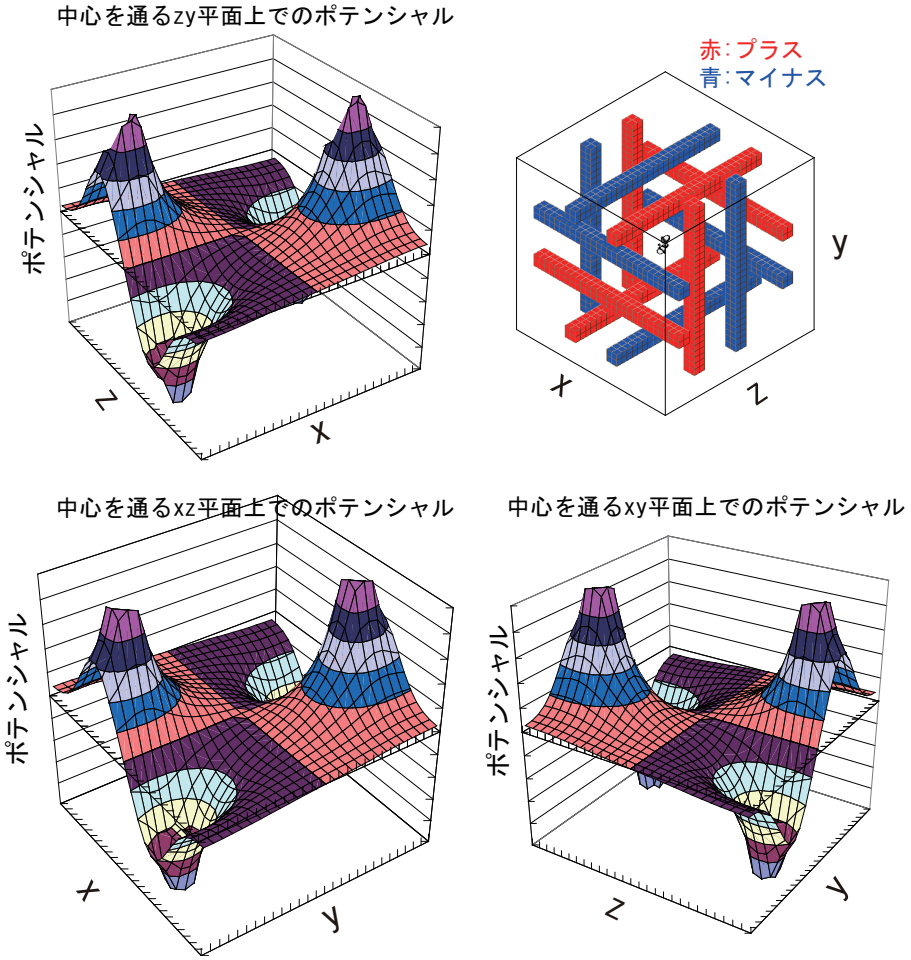



図 4.5: 中心を通る xy, zy, xz 平面におけるポテンシャル曲線.

xy, zy, xz 平面のいずれから見ても中心付近に鞍型ポテンシャルができていることがわかる．実際は交流電圧をかけているのでこれら 3 つの鞍型ポテンシャルは，振動しているので鞍点 (中心) にイオン (帯電液滴) をトラップできる．

## 4.3　シミュレーション結果

図 3.2 と図 3.3 を連結させたものが図 3.1 であり，以下にそのシミュレーション結果を示す．シミュレーション条件として Ion's Mass $2\times10^2$，Ion's Charge 1 Units, AMU_Mass_per_Charge $1\times10^2$，Ion's KE 0.1eV，Azimuth 0 Deg，Elevation 0 Deg, 空気抵抗に相当するダンピング定数 (Linear_Damping) は 0，$V_{ac}$=1kV，Ω=1MHz とした[*2)]．図 3.2, 図 3.3 のいずれの領域においてもイオンがトラップされているのが分かる．シミュレーション結果から, トラップ力は図 3.3 に対応する部分の方が強いことが分かる．　図 3.2 のセルではイオンは永年 (周期) 運動をしているので動いてはいるもののトラップできているといえる．

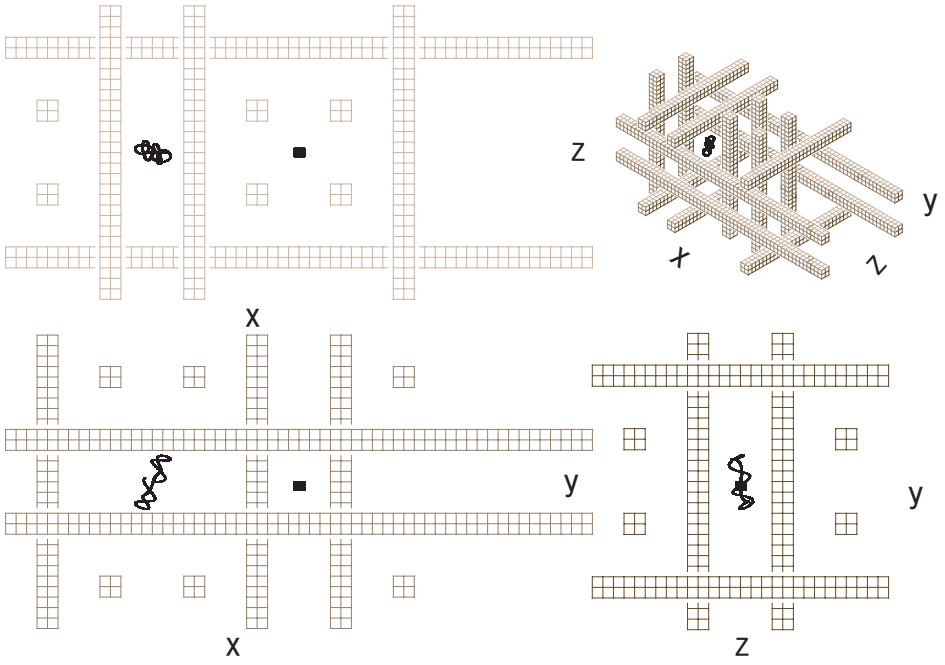



図 4.6: 図 3.1 のセルに対するシミュレーション結果.

SIMION の一番の特徴はユーザがプログラムを書くことができる点にある．このためプログラムを変えることで自由度が大きくなる．SIMION で用いるプログラムファイルは, 電極を作成したファイルが置いてあるフォルダ内に ”電極を作成した時と同じファイル名.PRG ”を作っておくと，シミュレーションする前に自動的にこのファイルを読みにいってくれる．実際にこの ”****.PRG ”にそれぞれの電極に交流電圧を設定するプログラムを書いたのでそれを付録 B に記す.

[*2)] 交流に関するデータ ($V_{ac}$ や Ω) はプログラムからしか設定できない.

図 4.7, 4.8 に実際に適当な交流電圧をかけてイオンをトラップできている様子を示す．見やすくするために空気抵抗に相当する (LinearDamping) は 0 とした．このパラメータに 0 でない値を入れると中心に収束していくのを確認した．

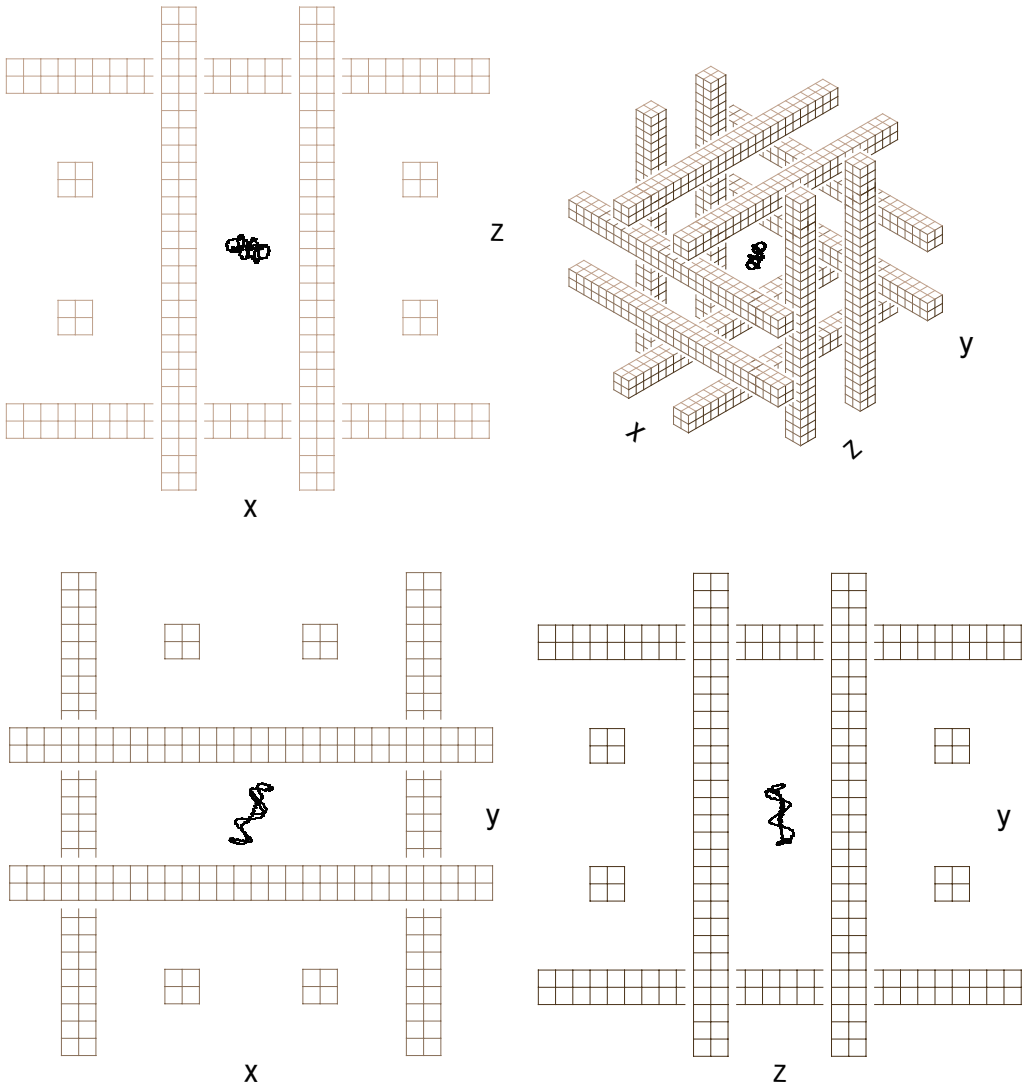



図 4.7: SIMION7.0 でイオンをトラップしている様子.
図 3.2 のセルに対するシミュレーション結果.

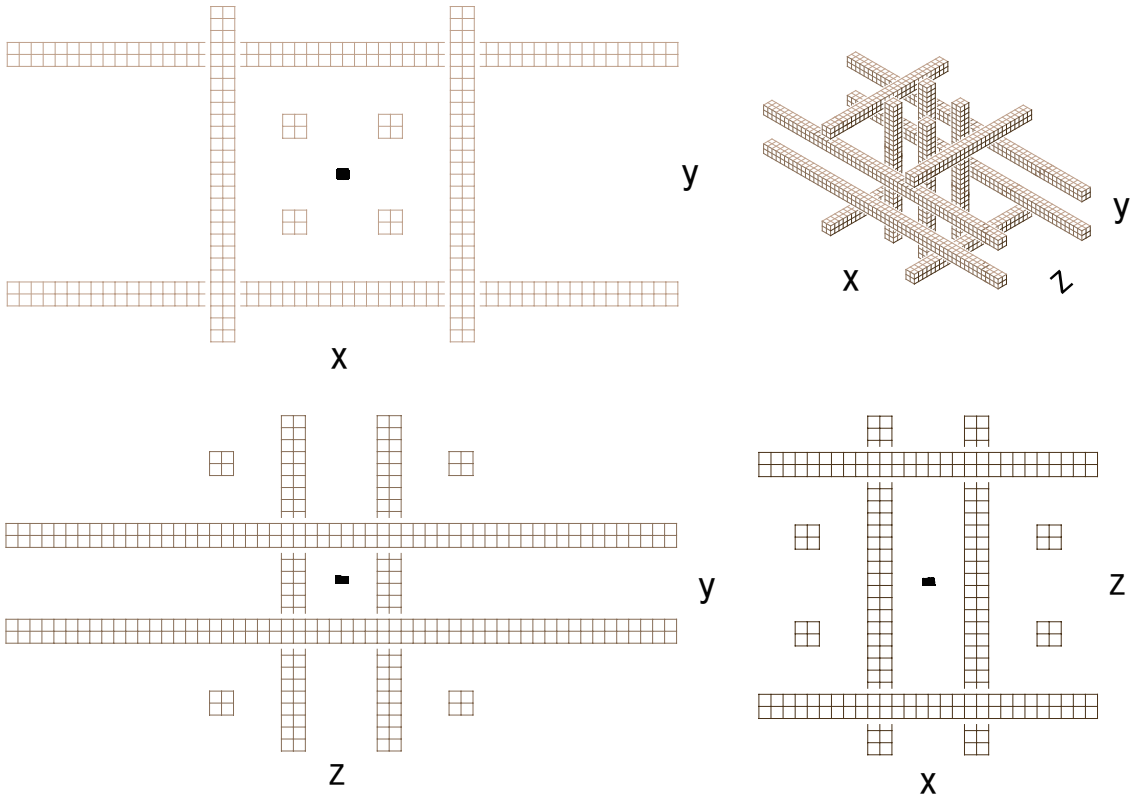



図 4.8: SIMION7.0 でイオンをトラップしている様子.
図 3.3 のセルに対するシミュレーション結果.

### 4.3.1 シミュレーションによる安定化領域

SIMION7.0 は $a$ パラメータ，$q$ パラメータの値を変えることができる．そして図 3.2, 4.7 に対する安定化領域ダイアグラムを求めたので図 4.9 に示す．本研究では $m/q$ を求めていないので，下記に示す適当な値でシミュレーションを行った．シミュレーション条件：

- Ion's Mass: $2 \times 10^2$ amu
- Ion's Charge: 1 Units
- Linear Damping: 0
- AMU_Mass_per_Charge: $2 \times 10^2$
- Ion's KE: 0.1eV
- Azimuth: 0 Deg
- Elevation: 0 Deg

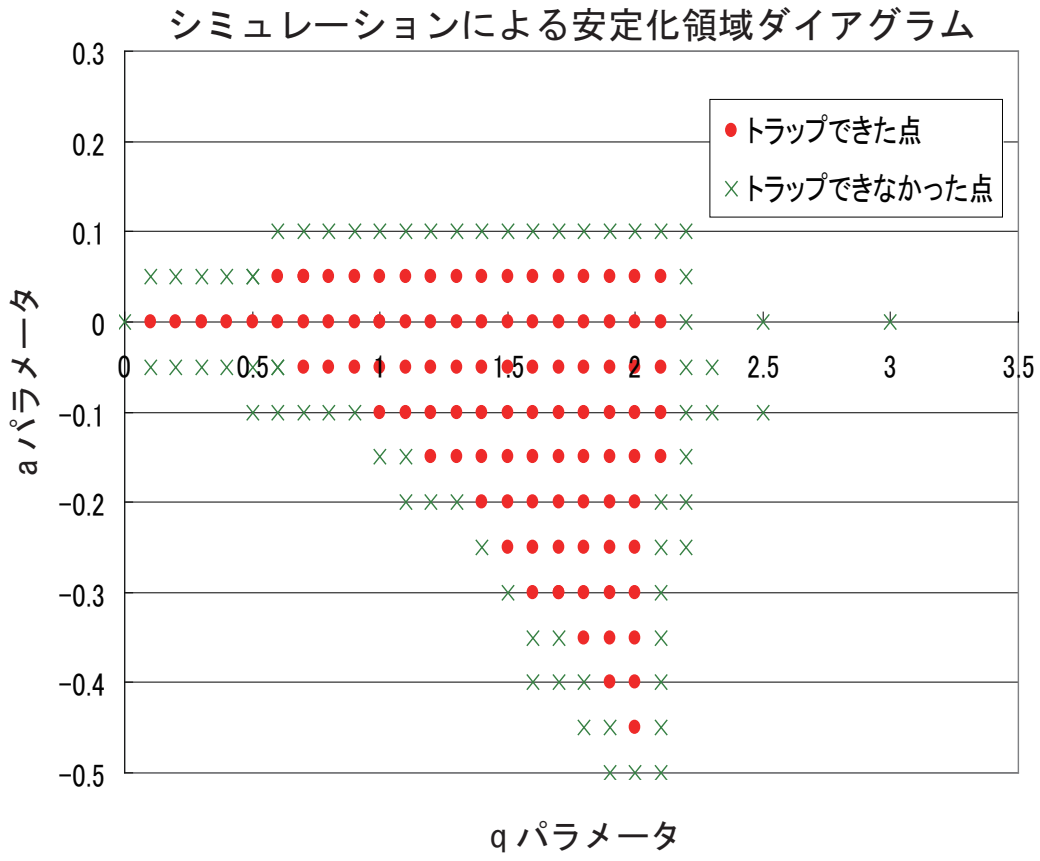



図 4.9: 図 3.2, 4.7 に対する安定化領域ダイアグラム (SIMION7.0 によるシミュレーション)．LinearDamping に値をいれれば更にトラップされる領域は広がる．

### SIMION8.0

2006 年 12 月 26 日に SIMION8.0.1 がリリースされた．

# 第5章　格子電極構造をもつ イオントラップの製作

アクリルキューブに設計図通り穴を空け，M3 ねじ棒で固定した．さらに除振台には M6 ねじで固定した[*1)]．最終的に完成した格子電極構造 (以下試作機 2 拡張版と呼ぶ) を以下に示す． 実際には図 5.3 に示すものを先に製作した．1cm×1cm×1cm のア

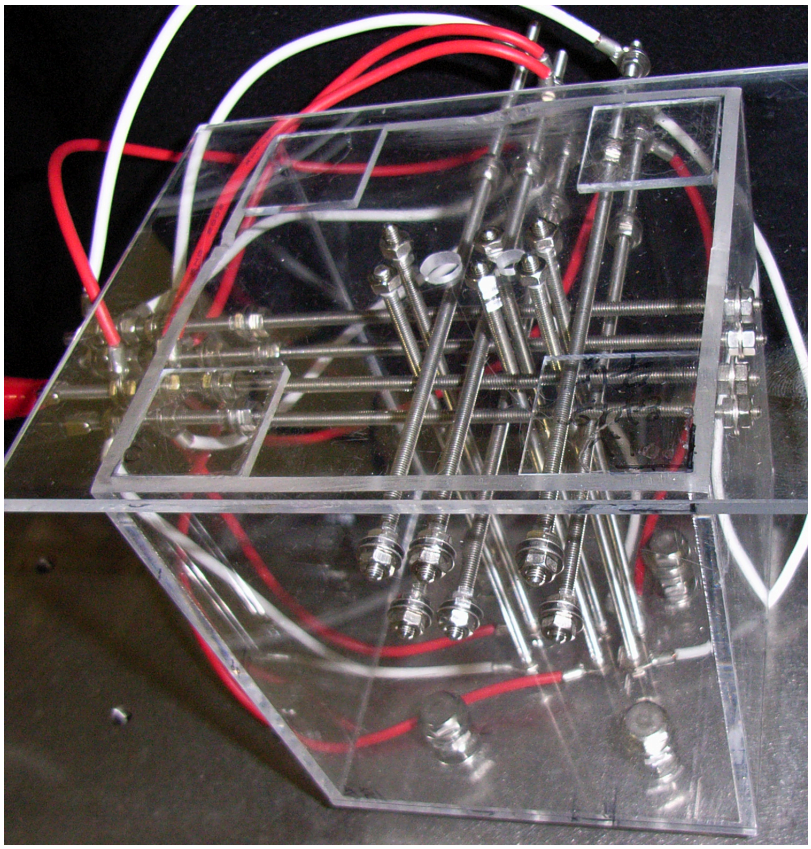


図 5.1: 格子電極構造の全体図．図 3.1 に対応．

[*1)] ねじを締めていく順番で苦労した．ふたの部分は内側から締めることができないので，締めすぎるとたわむので注意が必要である．

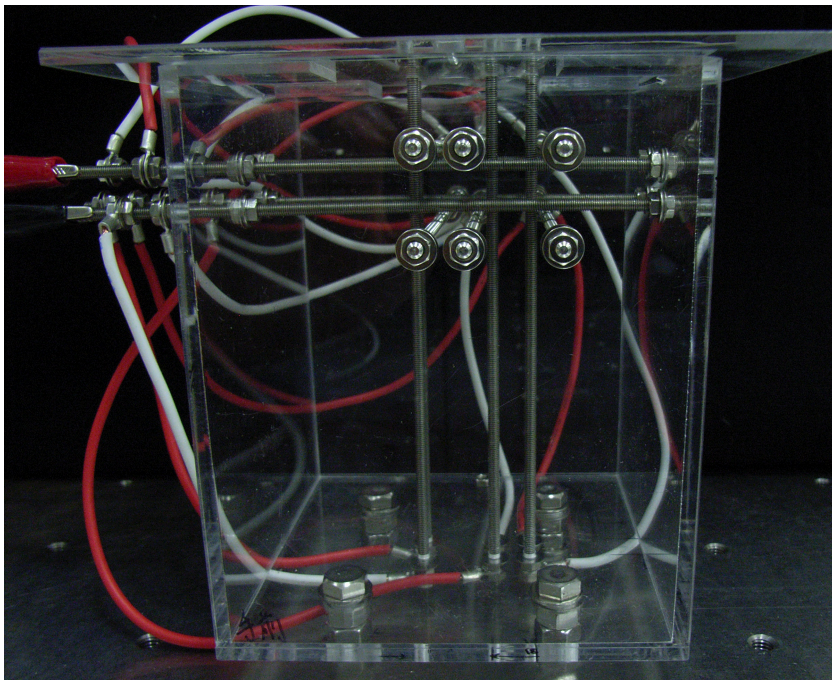


図 5.2: 格子電極構造の全体図を正面 (xy 平面) から見た図.

クリルキュービックに穴を開けて，M3 ねじ棒を固定した．作製した後に，この作製方法だと拡張する時に不便であることに気づき図 5.4 のタイプのものに変更した.

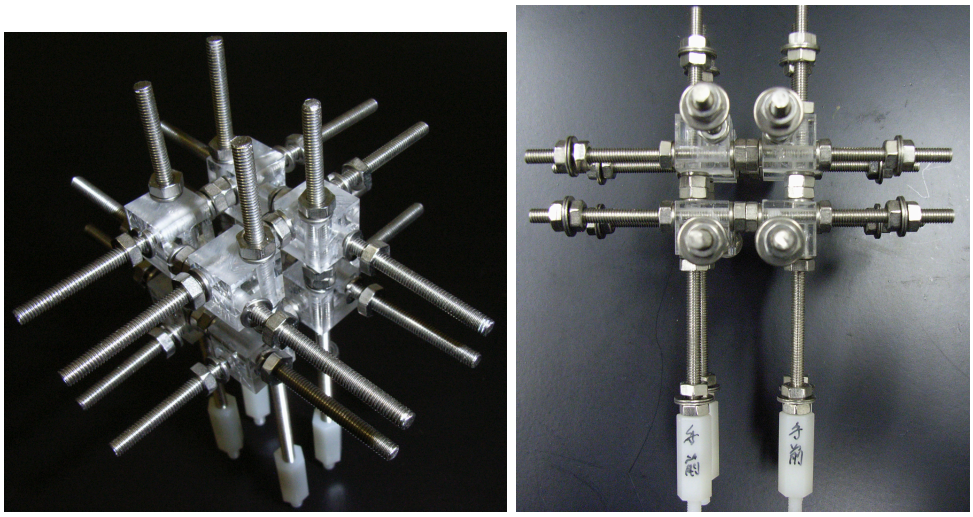


図 5.3: 試作機１．アクリルキュービックで M3 ねじ棒を固定した．

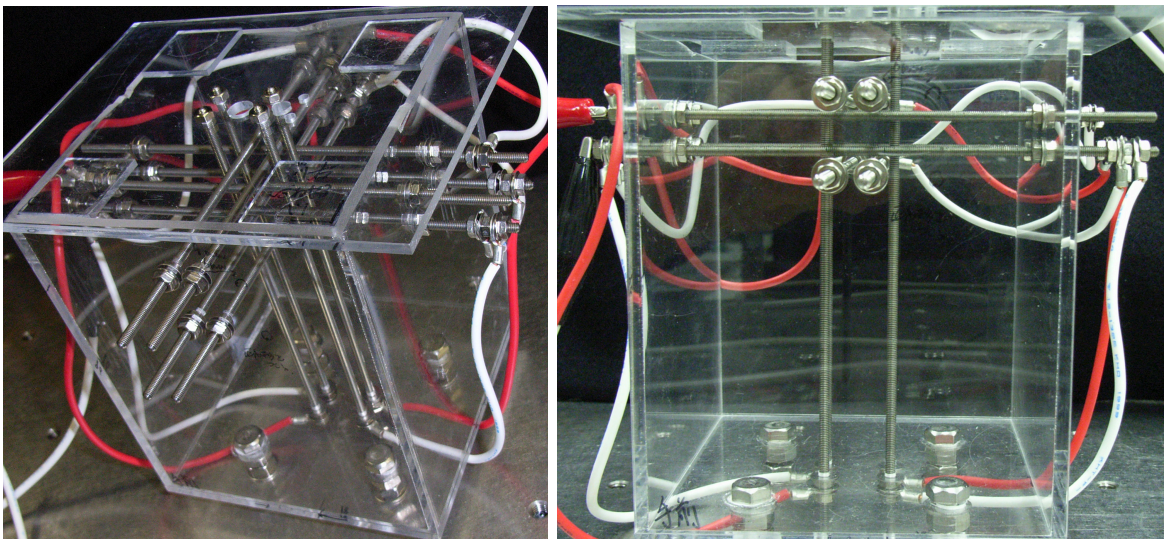


図 5.4: 試作機２．アクリルキューブに穴を開けて M3 ねじ棒を固定した．

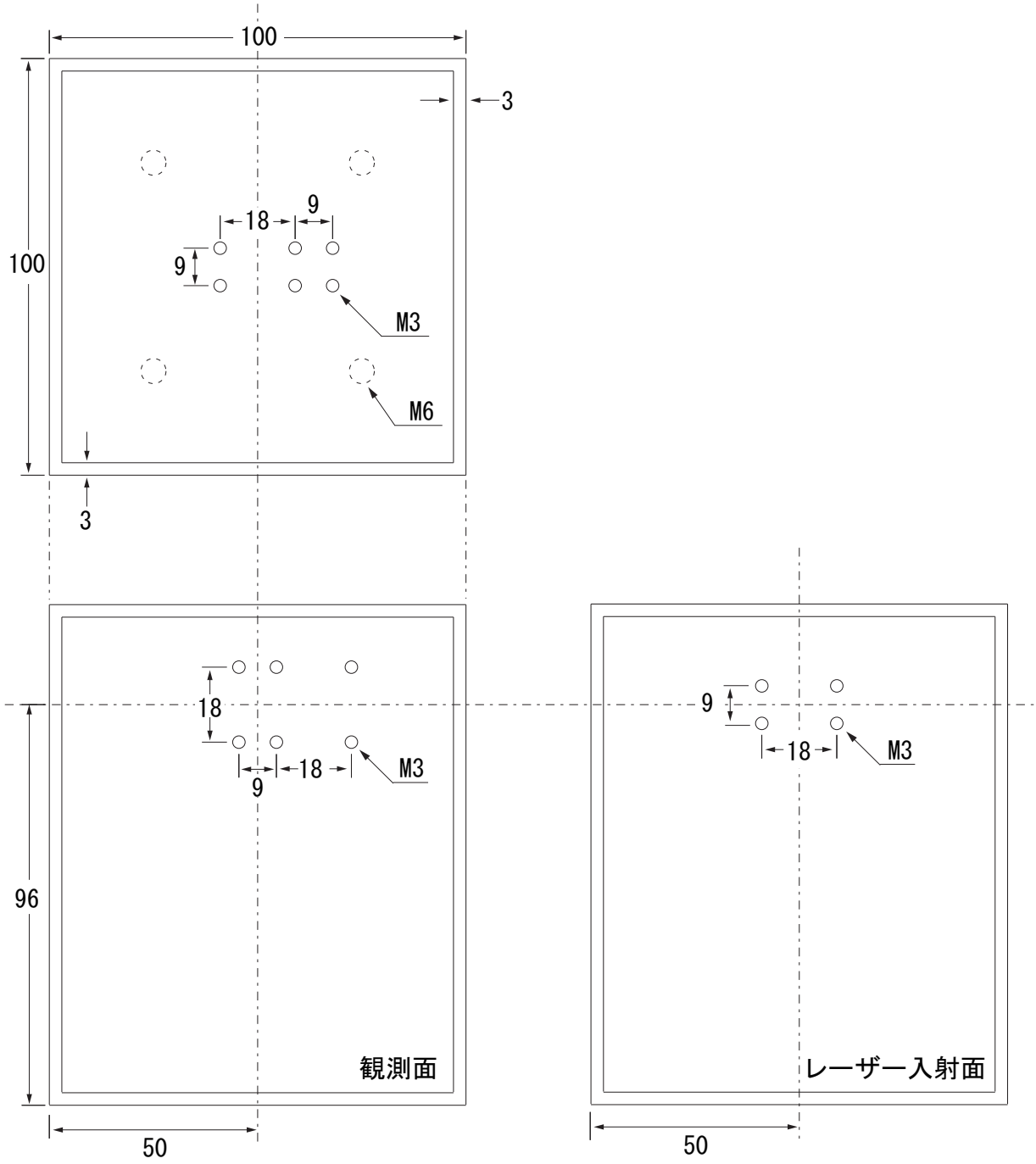



図 5.5: 格子電極構造を構成する M3 ねじ棒を固定するためのアクリルキューブ設計図．図 3.1, 4.6, 5.1, 5.2 に対応.

# 第6章　実験

## 6.1　実験装置の全体図

実験装置の全体図を図6.1に示す．注射針に2.5kV程度までの電圧をかけ帯電液滴を生成し，トラップ装置である格子電極構造部に落とす．格子電極には，1kVpp@300Hzの電圧を加えている．トラップした帯電液滴 (半径約10$\mu m$) はHe-Neレーザーで照射されることにより目視できるほどの光散乱を起こす．

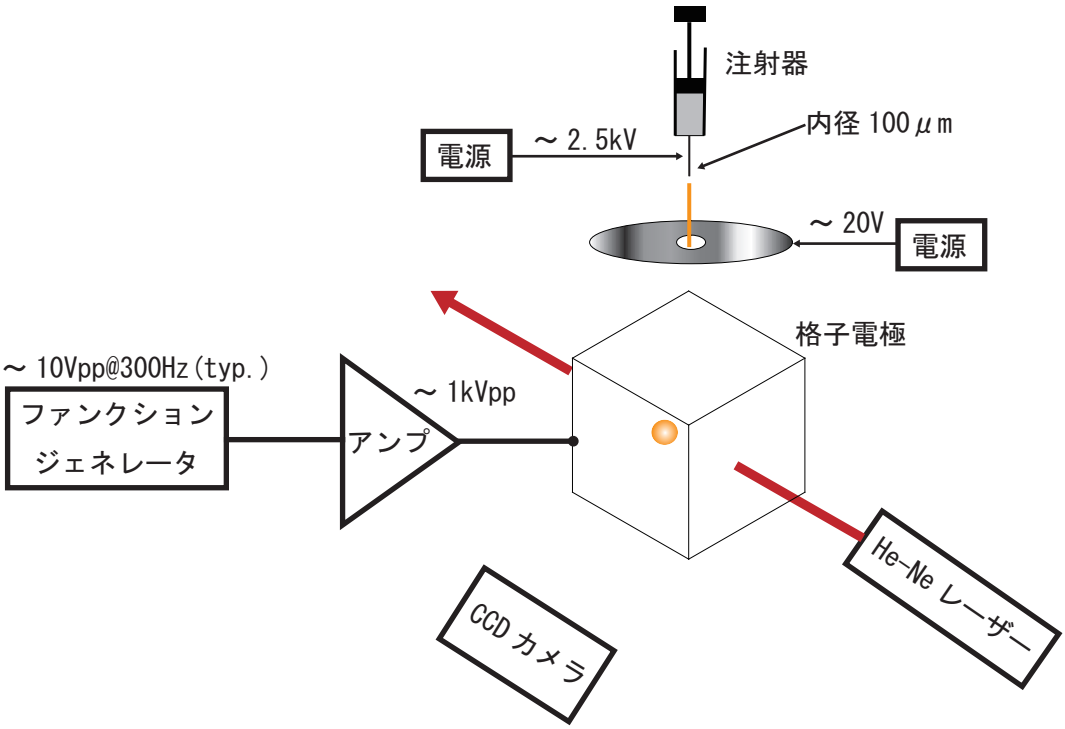



図 6.1: 実験装置の全体図.

ファンクションジェネレータ (HP 33120A, Agilent Technologies)[*1)]，アンプ (BOP 1000M POWER SUPPLY, KEPCO, Inc.) CCDカメラ (MINTRON MTV-6368N)[*2)]

[*1)]出力終端のデフォルト設定が50 Ωになっており，実際の実験系では開放端になっているためファンクションジェネレータの表示値の2倍が格子電極にかかっている．例えばファンクションジェネレータの表示値が10Vppだったら実際には20Vppが電極にかかっている．詳しくは取扱説明書参照.

## 6.2　帯電液滴のトラップ

### 6.2.1　試作機１でのトラップ結果

図5.3(試作機1)で実際にトラップできた様子が図6.2である．1kVpp@300Hz+Offset0Vでトラップしたが，周波数が307Hzですぐにトラップから外れてしまった．ねじ棒を固定している絶縁体のアクリルキュービック[*3)]が影響しているものと思われる．

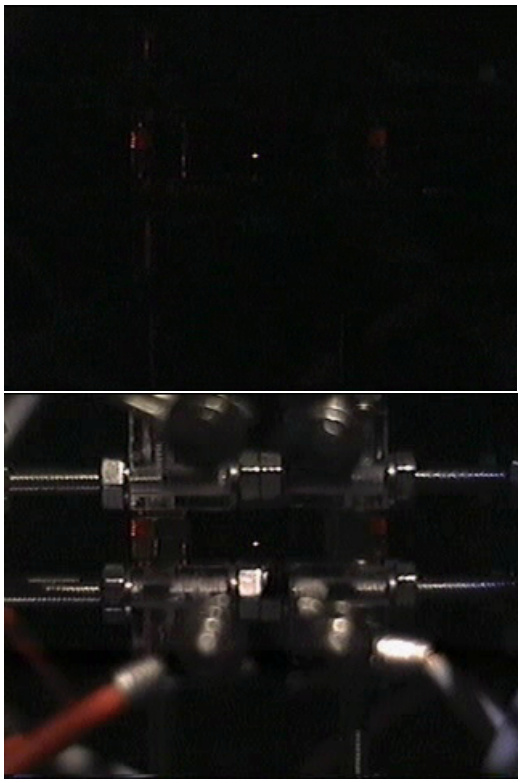


図 6.2: 試作機 1 においてトラップできた帯電液滴.
上と下の写真は同じ位置に対応している.

[*2)]IMAGE SENSOR: 1/3-inch CCD Image Sensor, CCD TOTAL PIXCELS: 811(H) × 508(V), SCANNING SYSTEM:525lines, 60fields/sec.
[*3)]絶縁物のため電位が決まらないので電場空間を乱す.

### 試作機 1 におけるリニアトラップ

当初予定していたところと違う所にトラップされた写真を以下の図 6.3 に示す．4 つ程トラップされているのが分かる．M3 のねじ棒を配線する時に使ったナットでねじ棒どうしの距離が縮まったことによりトラップできたと考えられる．

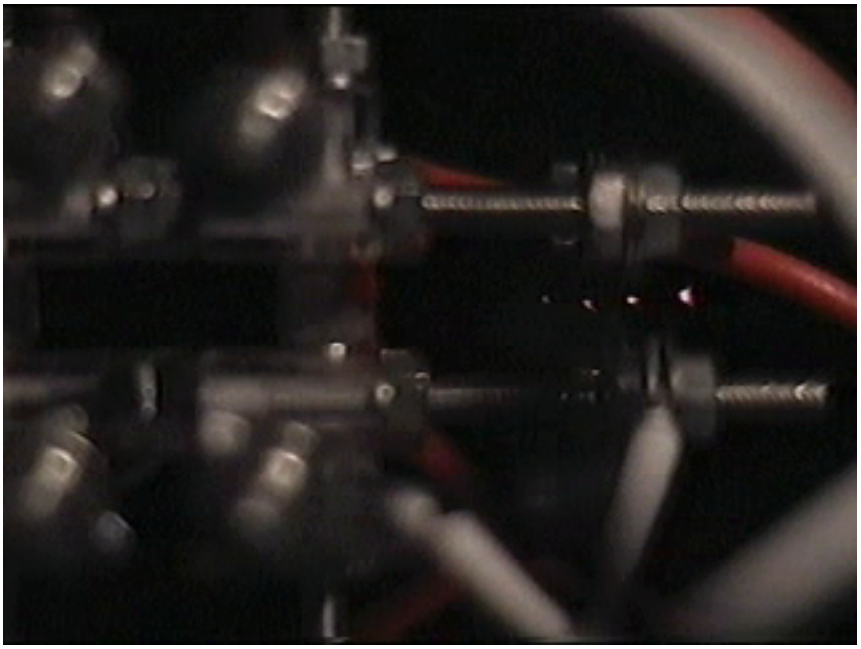


図 6.3: 試作機 1 において予定していたところと違うところにリニアトラップされた帯電液滴．

### 6.2.2　試作機２でのトラップ結果

図 5.4 に対応する試作機 2 でトラップできたときの様子を図 6.4 に示す．図 6.4(a) では安定にトラップされている．(b) は周波数を下げていった時に不安定になった時の写真である[*4)]．

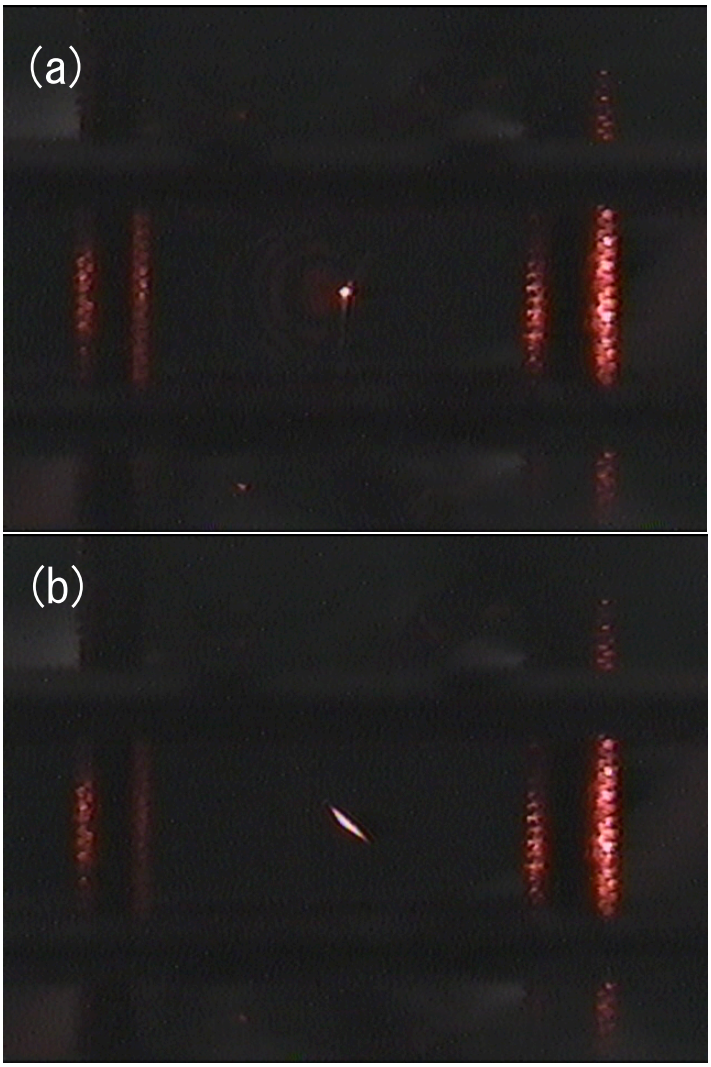



図 6.4: 試作機 2 においてトラップできた様子．(a) は安定してトラップされたとき，(b) は不安定にトラップされて振動している様子．(a) と (b) は同じ位置に対応している．

[*4)] 周波数が低い時には不安定になると振動し出したのが分かるが，周波数が高いときに不安定になると一瞬に飛んでいってしまうことが多い．一瞬に飛んでいってしまえば安定化領域は明確に線を引けるが，振動しているところも安定化領域の内に入るので実際の安定化領域の境界はぼやけていると見るべきだろうか．

## 6.3 試作機２に対する安定化領域ダイアグラム

式 (2.24), (2.25) から導かれた安定化領域ダイアグラム図 2.3 に対応するものが本研究においても得られたので図 6.5 に示す．ただし式 (2.24), (2.25) の $e, m, r_0, z_0$ は未知なので任意定数とし (式 6.1, 6.2)，適度な $a$ パラメータ, $q$ パラメータの数値にフィッティングした．安定化領域を示す黒の実線は青のひし形をもとに, 適当な関数でフィッティングした近似的なものである[*5)]．

$$a_z = -2a_r = A\frac{U}{\Omega^2} \tag{6.1}$$

$$q_z = -2q_r = Q\frac{V}{\Omega^2} \tag{6.2}$$

双曲面型であれば A=−2Q の関係があるが，図 6.5 では A と Q はそれぞれ別個な定数として扱った．

[*5)] 実験中 5.02Vpp と 2.52Vpp において出力リレーの切り替えにより, 出力電圧が瞬間的に 0V になり一瞬飛んでいきそうになり驚いた．詳細は取扱説明書を参照．

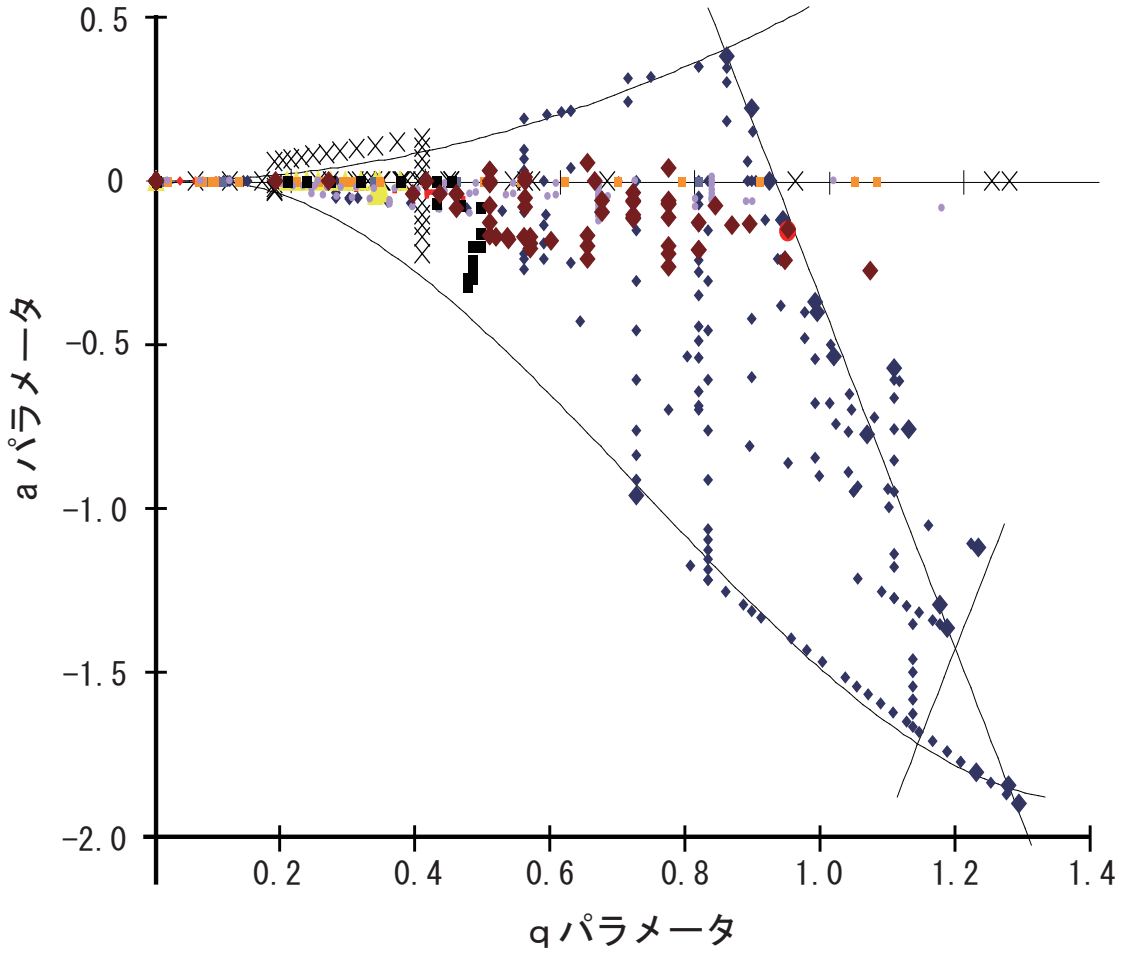



図 6.5: 試作機 2 における安定化領域ダイアグラム．測定点の色・模様が違うのは別々 ($m/q$ が違うと考えられる) の帯電液滴に対応する．大きいひし形は図 6.4 に代表されるようにイオン (帯電液滴) が振動してしまいトラップから逃げていきそうな点である．

## 6.4　試作機２拡張時のトラップ結果

図 6.6 に試作機 2 拡張時においてトラップできた写真を示す．はじめに右側の粒子をトラップして次に左側の粒子をトラップした．適当に周波数と電圧を調整することで意図的にそれぞれのセルに 1 個の帯電液滴がトラップされるようにした．He-Ne レーザーは写真右側から入射されており，トラップされた粒子がほぼ一直線上に並んでいるために同時に 2 つの粒子が照射されている．また，トラップした帯電液滴が振動し

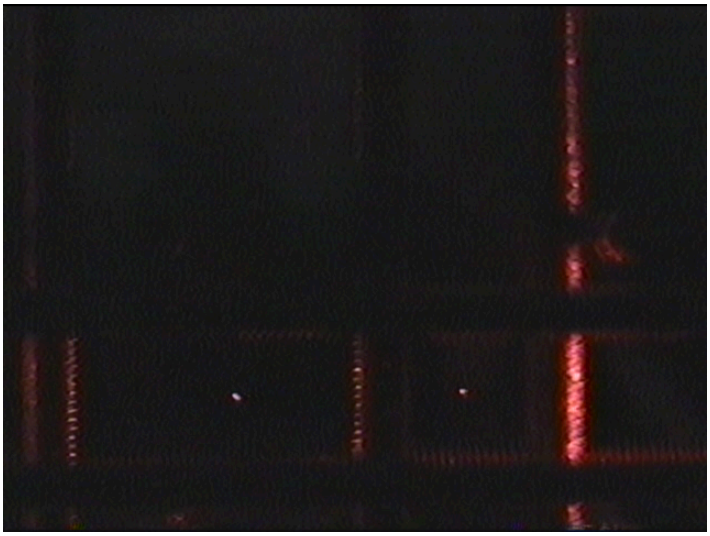


図 6.6: 試作機 2 の拡張版においてそれぞれのセルに 1 個ずつの液滴をトラップした様子．図 5.2 に対応

始める近傍の周波数で時間が経つと振動現象が起きることから，時間が経つことと周波数を下げることは同等であるということも分かった．式 2.24, 2.25 を見ると周波数を下げると $a$ パラメータ, $q$ パラメータ共に大きくなる．時間が経つとグリセリン液滴は蒸発していく，つまり質量が減る．この時いくらかの電荷も持って蒸発していくと考えられる．そして $a$ パラメータ,$q$ パラメータが大きくなるためには

$$\frac{\Delta e}{\Delta m} > 1 \tag{6.3}$$

ただし蒸発していくので $\Delta m < 0$ である．電荷が増えるというのは考えにくい，つまり $\Delta e < 0$ である．ここから分かることは電荷よりも質量の方が減り方が大きいということである[*6)]．

[*6)] 質量の減り方が大きいという事は極限を取ったら最後には電荷だけが残るのだろうか．あくまで今回の実験範囲において質量の減り方が電荷に比べ大きいと考えるべきだろう．

### 試作機 2 拡張時のリニアトラップ

試作機 2 拡張時の実験中，なかなかトラップされず本来重力で下に落ちていくはずの粒子が上に上がっていくので観察したところ，多くの帯電液滴がリニアトラップされていた．

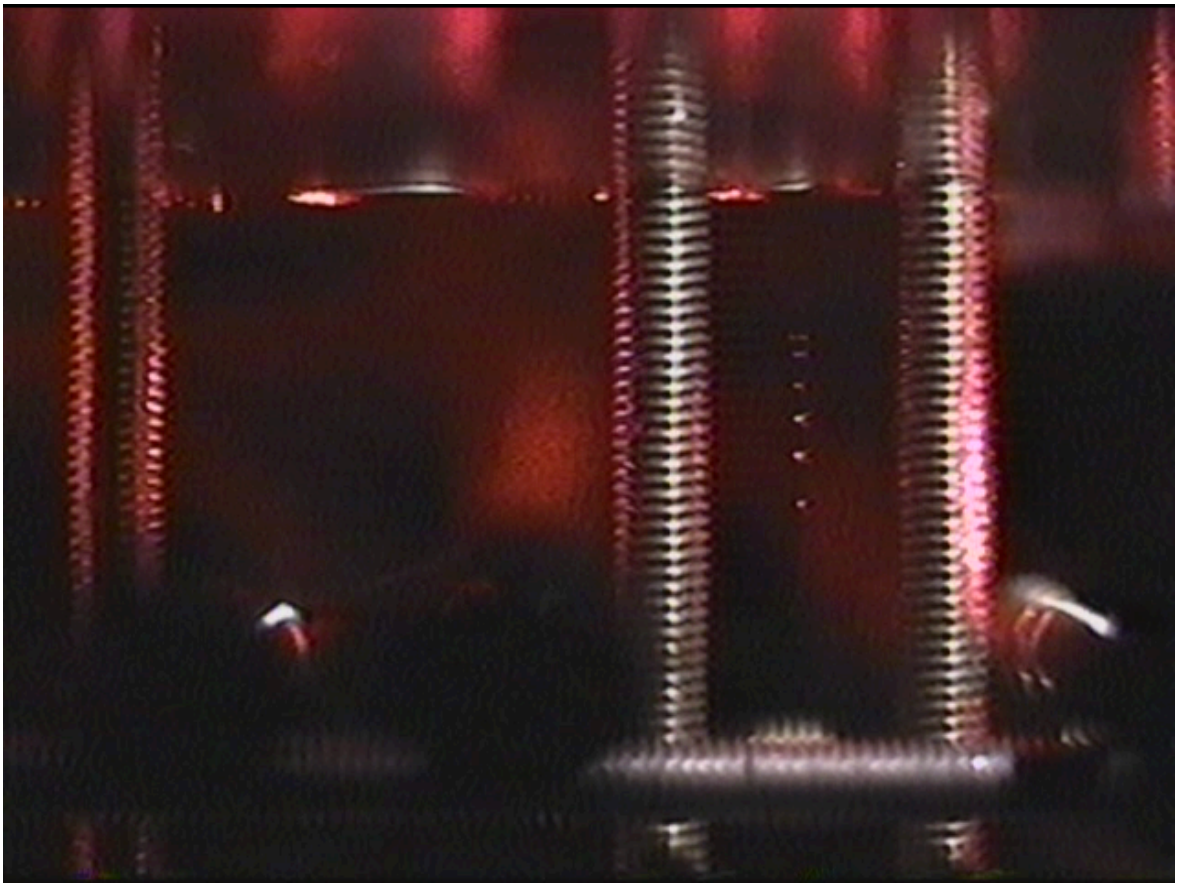


図 6.7: 試作機 2 の拡張時において右側のセル (図 3.3, 4.8 に対応) の上方においてリニアトラップされた多数の帯電液滴．

周波数を上げていくとこれらリニアトラップの粒子が安定になりきれいに一直線上に並ぶ．その結果クーロン反発力が強まり，図 6.6 の右側にトラップされている粒子が下に動くのを確認した．

### 試作機 2 拡張時のリニアトラップにおける永年運動

試作機 2 の右側のセルの上部でリニアトラップができた液滴を不安定にさせた時に永年運動[*7)]と呼ばれる運動を見ることができた．右側のセルの上部でリニアトラップできることはシミュレーションでも確認している．試作機 2 拡張時の左側のセルの上部では実験でもシミュレーションでもリニアトラップはできなかった．

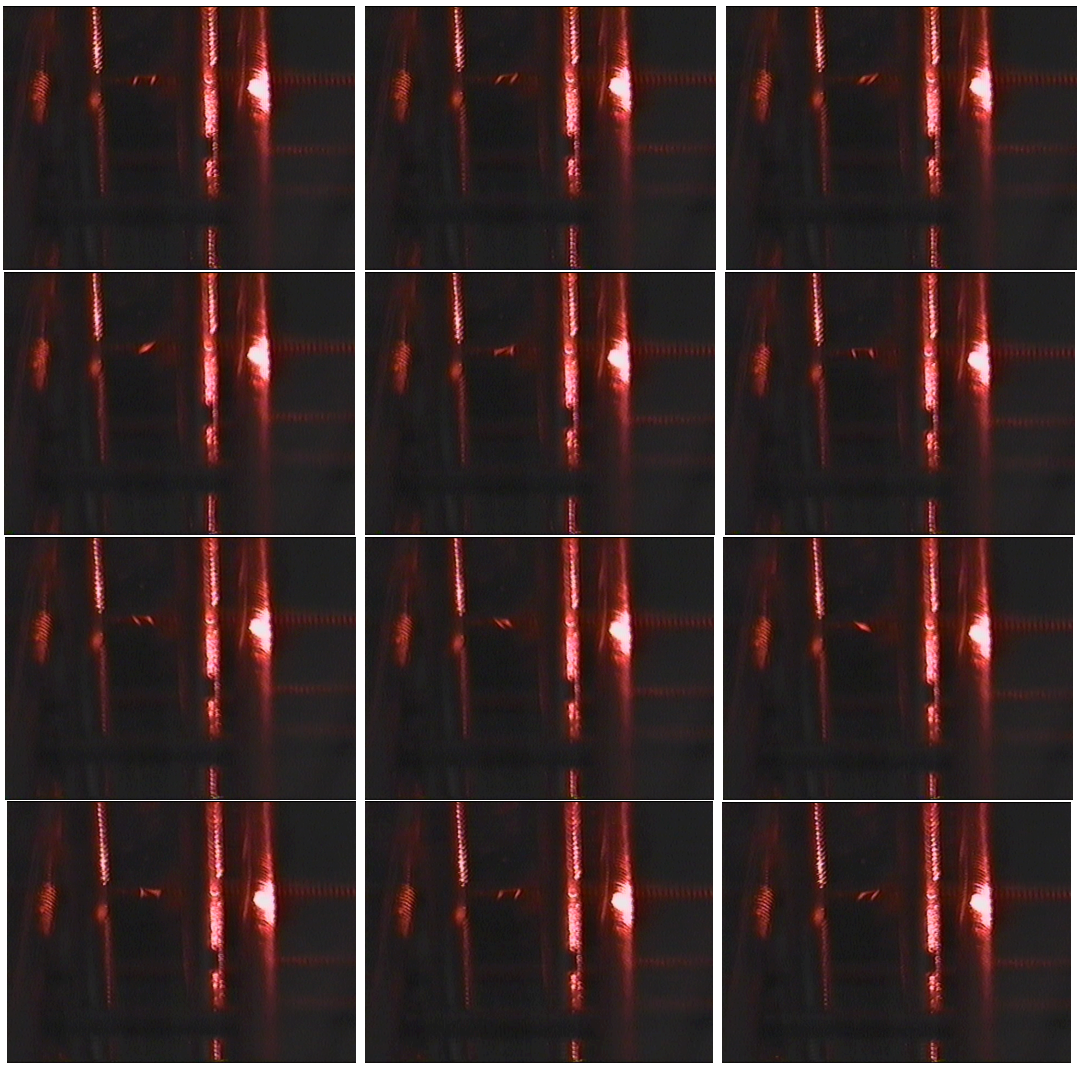


図 6.8: 1/30 秒ごとの 12 枚の写真．永年運動をしている．左上から順に右に見ていくと 12/30=0.4$sec$ が一周期であることが分かる．斜め上方から撮影．

[*7)] 英語で secular motion.

### 6.4.1 トラップした帯電液滴による光散乱

図 6.6 の右側のトラップ粒子に対して He-Ne レーザー (偏光方向:垂直) を照射した時の光の散乱模様 (Mie 散乱 [11–15]，付録 C 参照) 図 6.9 を見ることができた [16].

#### 最小二乗法によるトラップした帯電液滴のサイズ決定

図 6.9 の散乱模様の輝度 (特に赤に関して) を数値化して計算値と二乗和をとったのが図 6.10 である．そして二乗和の最も小さい (最小二乗法) 半径を候補値として 9.710$\mu$m と 9.880$\mu$m を図 6.11 に載せた．どちらも甲乙つけがたく似ている．今回とれたデータの縞の本数が少ないためこれ以上の限定は難しいものと思われる．なお，中心角度は本来レーザーに対して直角にカメラを置いているはずなので 90° になると思うかもしれないが，角度もある程度フィッティングにかけた．本研究ではサイズ測定を目的としていなかったため，カメラの設置角度を正確に決めるシステムがなかったためこうなった[*8)].

[*8)] フィッティングには mathematica5.2 を用いた．NMinimize など便利なコマンドがある．また画像データを数値化することもできるが，国立天文台のホームページ http://www.nao.ac.jp/index.html に便利な画像解析ソフトがある．

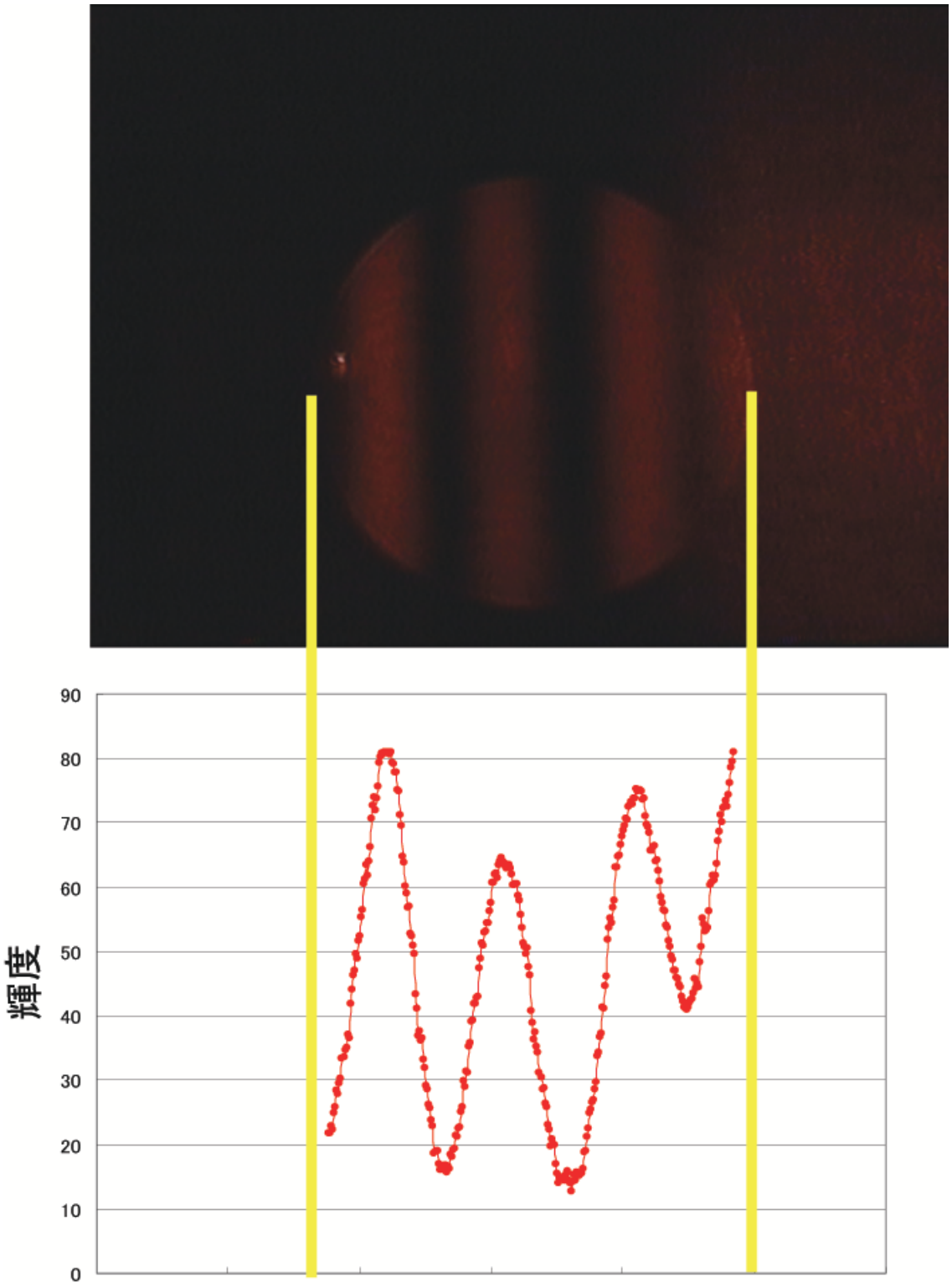



図 6.9: トラップした粒子によるレーザー光の散乱パターン．右側のざらついた雑音は電極棒からの散乱．一番左の縞の横にある輝点は，トラップした粒子ではなくアクリル箱についているほこりである．

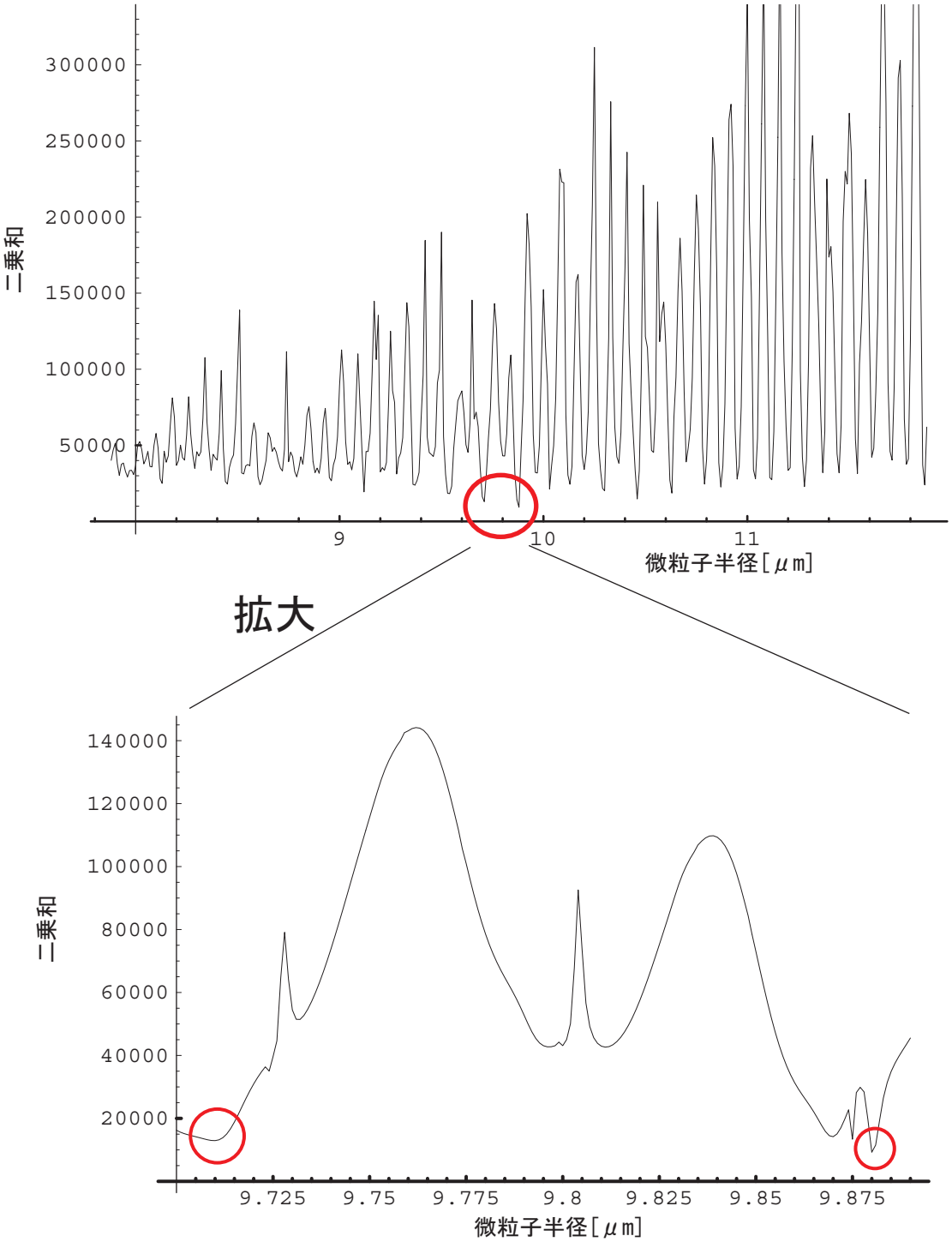
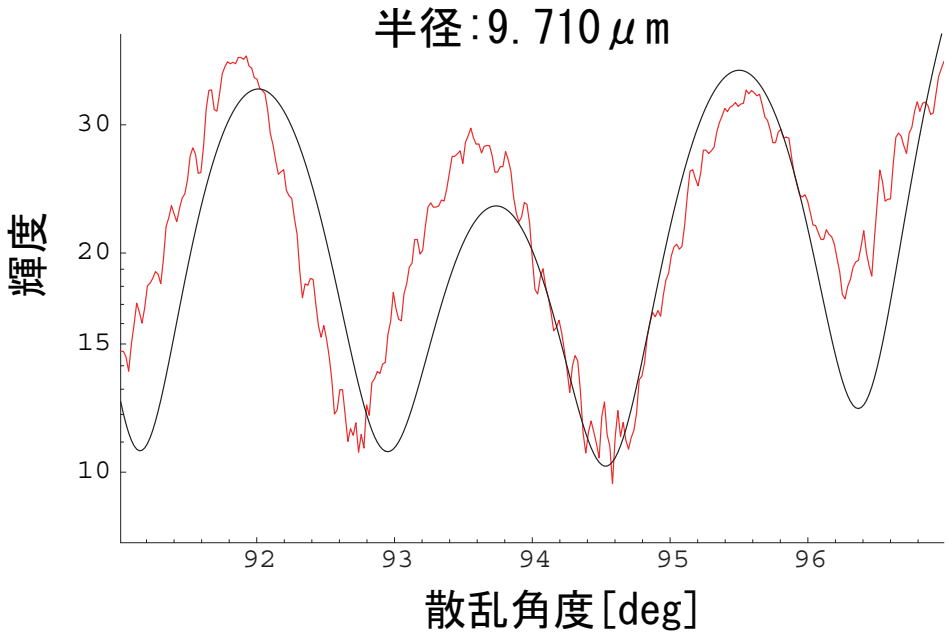



図 6.10: 半径を横軸にとり，図 6.9 のデータを計算値と比較し二乗和をとった．上が 10nm 刻みで約 8$\mu$m〜12$\mu$m．下が 9.7$\mu$m〜9.9$\mu$m までを 1nm 刻みで計算した．候補値 9.710$\mu$m と 9.880$\mu$m を得る．



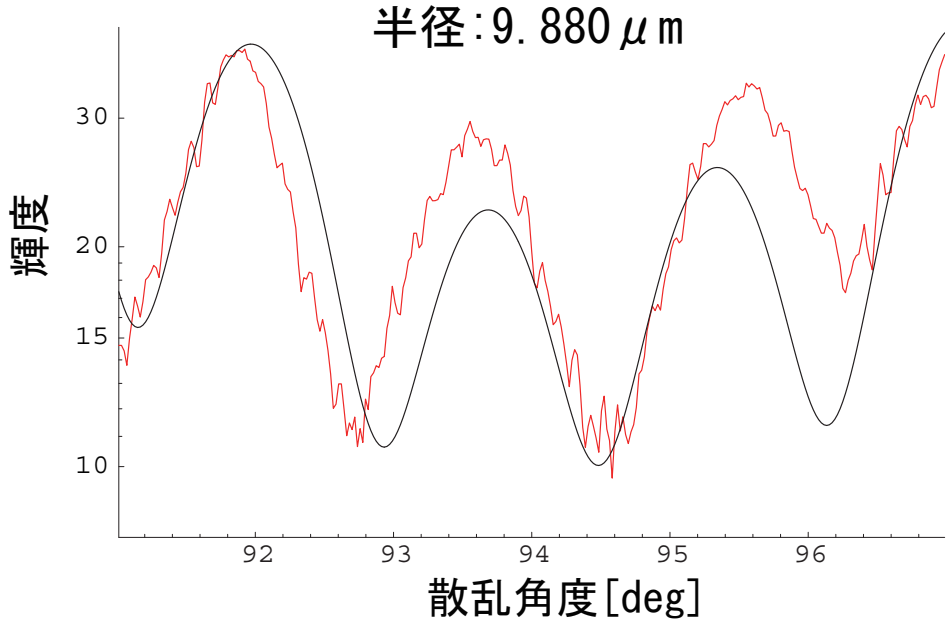




図 6.11: 図 6.10 のデータから候補値 9.710$\mu$m, 9.880$\mu$m を算出．それぞれの半径において定数倍およびオフセットを最適化したあとの図．どちらも似ており甲乙つけがたい．二乗和を比べると 9.710$\mu$m の方が小さい．

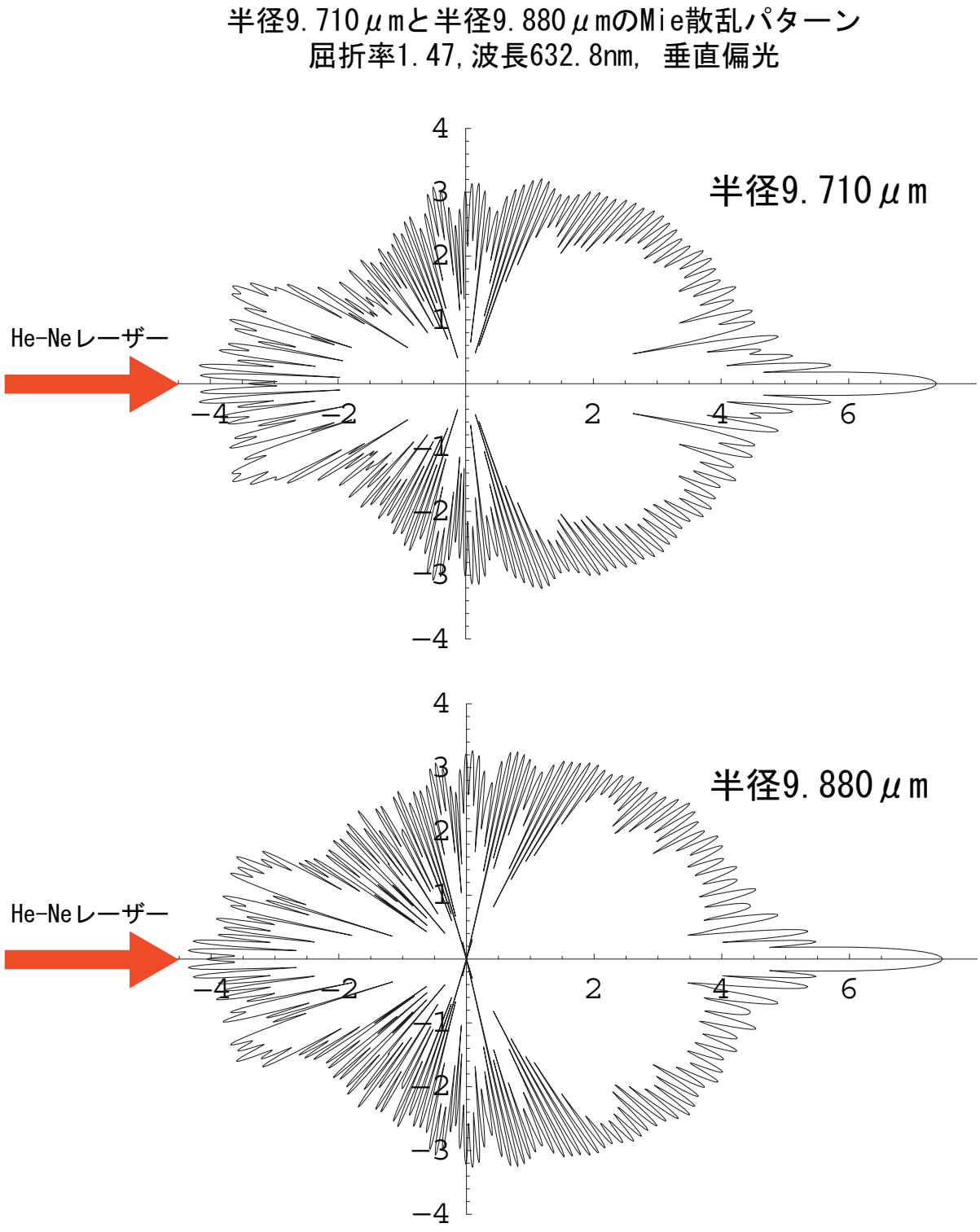



図 6.12: 候補に挙がった 9.710$\mu$m と 9.880$\mu$m の Mie 散乱パターン計算値. 図 6.11 の全体図．Log スケールで書いている．屈折率の虚部, すなわち吸収は 0 としている.

**図 6.6 の左側のトラップ粒子に対して He-Ne レーザー (偏光方向:垂直) を照射した時の光の散乱模様**

試作機 2 拡張時の右側のセルにトラップされた粒子 (図 6.9) の時のような縞模様が見えない．おそらくこの画面では見えない範囲で微小に振動していることにより，干渉がきれいに起こっていないためと考えている．

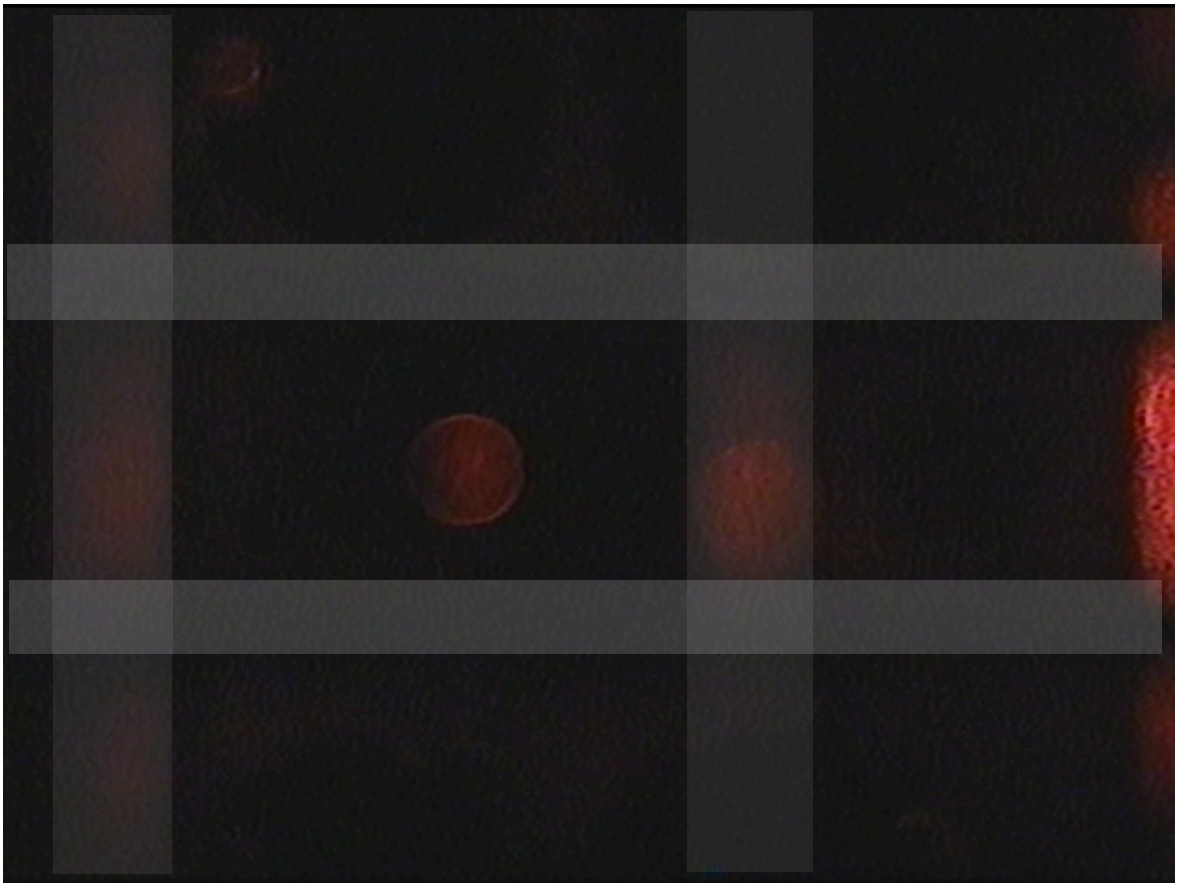


図 6.13: 図 6.6 の左側にトラップした粒子は，なぜか図 6.9 のように光の散乱による縞模様は見えなかった．白い透明部分は電極部を表現したもの．

# 第7章　結論

本論文において3次元的に拡張できる格子電極構造を示し，イオン (帯電液滴) を捕捉できることをシミュレーション及び実験で確かめた．今回提案した格子電極構造が双曲面型と同じような安定化領域を持っていることも確認できた．本研究では容易に手に入るM3ねじ棒を使用したので，簡便でありいろいろと試行錯誤ができる．原子イオンのトラップには超高真空装置が必要であるが，本実験は大気圧中でトラップしているためにそのような装置もいらない．複雑な電極構造をしていないために観測がはるかに容易で，三次元的にトラップした微粒子を見ることができる．本実験で扱った液滴の質量は原子イオンのそれに比べてはるかに大きいが，イオントラップ中での粒子の振る舞いは粒子の質量，電荷，交流電圧と周波数，トラップ空間で表されるパラメータで決まるため，本実験の結果を原子イオンのトラップにスケーリングすることができる．

# 付 録A　Mathieu方程式に関するプログラム

```
1  Off[General::spell];(*似ているスペルでも警告文を表示させない*)
2  (*初期値設定*)
3  ar = 0;
4  qr = 0.06;
5  az = 0;
6  qz = 0.07;
7  tstart = 0(*解析開始時刻見たい時間帯を入力*);
8  tend = 1000(*解析終了時刻*);
9  delta = 1;
10 (*配列が [[1]] からスタートすることから派生する+1*);
11 numlis = (tend - tstart)/delta + 1
12 (*r 方向の数値解析*)
13 solutionr = NDSolve[
14         {
15         ur"[t] + (ar - 2*qr*Cos[2*t])*ur[t] == 0
16         , ur '[0] == 0.1        (*初期条件 tstart と混同してはいけない*)
17         , ur[0] == 0            (*初期条件 tstart と混同してはいけない*)
18         }
19         , {ur}                  (*未知関数*)
20         , {t, tstart, tend}     (*変数の範囲*)
21 (*振動しているので初期設定 (数値が小さい 3000 程度) のままだとエラーが出る*)
22         , MaxSteps -> 10000
23         ];(*セミコロンをつければコメントによるバグが防げる*)
24 Plot[
25         (*solutionr を一時的に ur 関数に代入*)
26         Evaluate[ur[t] /. solutionr]
27         , {t, tstart, tend}(*この tstart は下のプログラムに影響を与えない.*)
28         , PlotLabel -> StyleForm["r 方向 vs 時間", FontSize -> 13]
29 ];
30  (*z 方向の数値解析*)
31  solutionz = NDSolve[
32         {
33         uz"[t] + (az - 2*qz*Cos[2*t])*uz[t] == 0
34         ,uz"[0] == 0.1          (*初期条件 tstart と混同してはいけない*)
35         , uz[0] == 0            (*初期条件 tstart と混同してはいけない*)
36         }
37         , {uz}                  (*未知関数*)
38         , {t, tstart,tend}      (*変数の範囲*)
39 (*振動しているので初期設定 (数値が小さい 3000 程度) のままだとエラーが出る*)
40         , MaxSteps -> 10000
```

```
41          ];(*セミコロンをつければコメントによるバグが防げる*)
42  Plot[
43          Evaluate[uz[t] /. solutionz](*solutionz を一時的に uz 関数に代入*)
44          , {t, tstart, tend}(*この tstart は下のプログラムに影響を与えない.*)
45          , PlotLabel -> StyleForm["z 方向 vs 時間", FontSize -> 13]
46    ];
47  (*数値解析結果のリスト処理*)
48  (*数値解析結果は御親切にもそれぞれの結果に{}をつけてくれているので,
49  {}を取り除く作業に一番時間がかかっている*)
50          rr =
51          Table[
52          Part[
53          Table[
54          ur[t] /. solutionr
55          , {t, tstart, tend,
56         delta}                    (*0 から tend まで 1 刻みで*)
57          ]
58          , t, 1   (*{}を取り除くためのコマンド文:t 番目の配列の 1 つ目の値*)
59          ]
60          , {t, 1, numlis}
61          ];
62          zz =
63          Table[
64          Part[
65          Table[
66          uz[t] /. solutionz
67          , {t, tstart, tend, delta}        (*0 から tend まで 1 刻みで*)
68          ]
69          , t, 1 (*{}を取り除くためのコマンド文:t 番目の配列の 1 つ目の値*)
70          ]
71          , {t, 1, numlis}
72          ];
73   zzvsrr = Table[{rr[[t]], zz[[t]]}, {t, 1, numlis}];
74   ListPlot[
75     zzvsrr
76     , PlotJoined -> True
77     , PlotLabel -> StyleForm["z 方向 vs r 方向", FontSize -> 13]
78     , AspectRatio -> Automatic
79     ];
```

# 付 録B　SIMIONで使用したプログラム

## B.1　交流電圧設定プログラム

```
1   ; Program for Novel 3D-Arrayed Ion Trap
2   ; with Electrodes of Lattice Structure
3   ; Yugo Onoda 2007
4
5
6   ; definition of user adjustable variables  --------------------
7
8           ; ---------- adjustable during flight -----------------
9
10  defa _Linear_Damping           1.0    ; adjustable variable for linear damping
11  defa _Qz_tune                  0.6    ; Qz tuning point
12  defa _Az_tune                  0.00   ; Az tuning point
13  defa _AMU_Mass_per_Charge    100.0    ; mass tune point in amu/unit charge
14  defa _Target_Voltage           0.0    ; voltage of sims target
15  defa _Left_Cap_Voltage         0.0    ; voltage on left cap
16  defa _Right_Cap_Voltage        0.0    ; voltage on right cap
17
18          ; ---------- adjustable at beginning of flight -----------------
19
20  defa PE_Update_each_usec       0.05  ; pe surface update time step in usec
21  defa Percent_Energy_Variation 90.0   ; (+- 90%) random energy variation
22  defa Cone_Angle_Off_Vel_Axis 180.0   ; (+- 180 deg) cone angle - sphere
23  defa Random_Offset_mm          0.1   ; del start position (x,y,z) in mm
24  defa Random_TOB                0.909091; random time of birth over one cysle
25
26  defa Phaze_Angle_Deg           0.0    ; entry phase angle of ion
27  defa Freqency_Hz               1.1E6  ; rf frequency of quad in (hz)
28  defa Effective_Radius_in_cm    0.41   ; effective quad radius r0 in cm
29  defa mm_per_Grid_Unit          0.1    ; grid scaling mm/grid unit
30
31
32
33  ; definition of static variables ------------------------------
34
35  defs first                     0.0    ; first call flag
36  defs scaled_rf                 0.0    ; scaled rf base
37  defs rfvolts                1000.0    ; rf voltage
```

```
38  defs dcvolts                  0.0    ; dc voltage
39  defs omega                    6.2    ; freq in radians / usec
40  defs theta                    0.0    ; phase offset in radians
41  defs reverse                 -1.0
42  defs Next_PE_Update           0.0    ; next time to update pe surface
43
44
45  ; program segments below ------------------------------------------
46
47
48  ;-------------------------------------------------------------------
49  ;seg initialize      ; randomize ion's position, ke, and direction
50          ;1 sto Rerun_Flym               ; force rerun on
51                                          ; turns traj file saving off
52          ;------------- get ion's initial velocity components -------
53          ;rcl ion_vz_mm                  ; get ion's specified velocity components
54          ;rcl ion_vy_mm
55          ;rcl ion_vx_mm
56
57          ;------------------- convert to 3d polar coords -------------
58          ;>p3d                            ; convert to polar 3d
59
60          ;------------------- save polar coord values ----------------
61          ;sto speed rlup                  ; store ion's speed
62          ;sto az_angle rlup               ; store ion's az angle
63          ;sto el_angle                    ; store ion's el angle
64
65          ;--------- make sure Percent_Energy_Variation is legal -------
66                  ; force 0 <= Percent_Energy_Variation <= 100
67          ;rcl Percent_Energy_Variation abs
68          ;100 x>y rlup sto Percent_Energy_Variation
69
70          ;-------- make sure Cone_Angle_Off_Vel_Axis is legal ---------
71                  ; force 0 <= Cone_Angle_Off_Vel_Axis <= 180
72          ;rcl Cone_Angle_Off_Vel_Axis abs
73          ;180 x>y rlup sto Cone_Angle_Off_Vel_Axis
74
75          ; ------------------- calculate ion's defined ke -------------
76          ;rcl ion_mass                    ; get ion's mass
77          ;rcl speed                       ; recall its total speed
78          ;>ke                             ; convert speed to kinetic energy
79          ;sto kinetic_energy              ; save ion's defined kinetic energy
80
81          ; -------------------- compute new randomized ke -------------
82                  ; convert from percent to fraction
83          ;rcl Percent_Energy_Variation 100 /
84          ;sto del_energy 2 * rand *       ; fac = 2 * del_energy * rand
85          ;rcl del_energy - 1 +            ; fac += 1 - del_energy
86          ;rcl kinetic_energy *            ; new ke = fac * ke
87
88          ; ---------------------- convert new ke to new speed -----------
89          ;rcl ion_mass                    ; recall ion mass
```

```
90          ;x><y                          ; swap x any y
91          ;>spd                          ; convert to speed
92          ;sto speed                     ; save new speed
93
94          ;-- compute randomized el angle change 90 -------
95          ;-- +- Cone_Angle_Off_Vel_Axis            -------
96          ;-------- we assume elevation of 90 degrees for mean ----------
97          ;-------- so cone can be generated via rotating az +- 90 -------
98                                     ; (2 * Cone_Angle_Off_Vel_Axis * rand)
99          ;2 rcl Cone_Angle_Off_Vel_Axis * rand *
100                                    ;  - Cone_Angle_Off_Vel_Axis + 90
101         ;rcl Cone_Angle_Off_Vel_Axis - 90 +
102
103         ;-------------- compute randomized az angle change ------------
104         ;--- this gives 360 effective because of +- elevation angels ---
105         ;180 rand * 90 -                ;          +- 90 randomized az
106
107         ;---------------------- recall new ion speed ------------------
108         ;rcl speed                     ; recall new speed
109
110         ;------- at this point x = speed, y = az, z = el --------------
111         ;--------- convert to rectangular velocity components ---------
112         ;>r3d                          ; convert polar 3d to rect 3d
113
114         ;------------- el rotate back to from 90 vertical -------------
115         ;-90 >elr
116
117         ;------------ el rotate back to starting elevation -------------
118         ;rcl el_angle >elr
119
120         ;------------- az rotate back to starting azimuth -------------
121         ;rcl az_angle >azr
122
123         ;------ update ion's velocity components with new values --------
124         ;sto ion_vx_mm                 ; return vx
125         ;rlup
126         ;sto ion_vy_mm                 ; return vy
127         ;rlup
128         ;sto ion_vz_mm                 ; return vz
129
130         ;--------- randomize ion's position components --------
131         ;rcl Random_Offset_mm
132         ;2 / sto half_pos              ; save half max shift
133
134         ;rcl ion_px_mm                 ; get nominal x start
135         ;rcl Random_Offset_mm rand * +  ; add random shift
136         ;rcl half_pos -                ; subtract half shift
137         ;sto ion_px_mm                 ; store random x start
138
139         ;rcl ion_py_mm                 ; get nominal y start
140         ;rcl Random_Offset_mm rand * +  ; add random shift
141         ;rcl half_pos -                ; subtract half shift
```

```
142         ;sto ion_py_mm                  ; store random y start
143
144         ;rcl ion_pz_mm                  ; get nominal z start
145         ;rcl Random_Offset_mm rand * +   ; add random shift
146         ;rcl half_pos -                  ; subtract half shift
147         ;sto ion_pz_mm                  ; store random z start
148
149         ;--------- randomize ion's time of birth --------
150         ;rcl Random_TOB abs rand *       ; create random time of birth
151         ;sto Ion_Time_of_Birth           ; use it for ion
152         ;------------------- done ----------------------------
153
154
155
156 ;-------------------------------------------------------------------------
157 seg Fast_Adjust                     ; generates trap rf with fast adjust
158                                     ; has first pass initialization
159     rcl first                       ; recall first pass flag
160     x=0 gsb init                    ; if this is first reference  --> init
161
162     rcl scaled_rf
163     rcl _AMU_Mass_per_Charge *      ; multiply by mass per unit charge
164     rcl _Qz_tune *                  ; rf tuning point
165     sto rfvolts                     ; save rf voltage
166
167     rcl scaled_rf
168     rcl _AMU_Mass_per_Charge *      ; multiply by mass per unit charge
169     rcl _Az_tune *                  ; substitute dc tune point
170     2 / chs                         ; additional dc factor
171     sto dcvolts                     ; save dc voltage
172
173 ;    rcl _Left_Cap_Voltage
174 ;    sto Adj_Elect01                 ; electrode 1 voltage
175 ;    rcl _Right_Cap_Voltage
176 ;    sto Adj_Elect03                 ; electrode 3 voltage
177 ;    rcl _Target_Voltage
178 ;    sto Adj_Elect04                 ; electrode 4 voltage
179
180     rcl Ion_Time_of_Flight          ; current tof in micro seconds
181     rcl omega *                     ; omega * tof
182     rcl theta +                     ; add phasing angle
183     sin                             ; sin(theta + (omgga * tof))
184     rcl rfvolts *                   ; times rf voltage
185     rcl dcvolts +                   ; add dc voltage
186     chs 2 * sto Adj_Elect01         ; electrode 1 voltage
187     chs 2 * sto Adj_Elect03         ; electrode 3 voltage
188 ;    chs 2 * sto Adj_Elect05         ; electrode 5 voltage
189
190     rcl Ion_Time_of_Flight          ; current tof in micro seconds
191     rcl omega *                     ; omega * tof
192     rcl theta +                     ; add phasing angle
193     sin                             ; sin(theta + (omgga * tof))
```

```
194       rcl rfvolts *                   ; times rf voltage
195       rcl reverse *
196       rcl dcvolts +                   ; add dc voltage
197       chs 2 * sto Adj_Elect02         ; electrode 2 voltage
198       chs 2 * sto Adj_Elect04         ; electrode 4 voltage
199   ;    chs 2 * sto Adj_Elect06         ; electrode 6 voltage
200
201       exit                            ; exit program segment
202
203
204   lbl init                            ; parameter initialization subroutine
205
206       1 sto first                     ; tunn off first pass flag
207
208       RCL Effective_Radius_in_cm      ; recall effective radius in cm
209       entr * 2 /                      ; (r * r)/2
210       rcl Freqency_Hz entr * *        ; multiply by frequency squared
211   ; -1.022442E-11 * Qz * MASS * FREQ * FREQ * R0 * R0
212       1.022442E-11 * chs
213       sto scaled_rf
214       rcl _AMU_Mass_per_Charge *      ; multiply by mass per unit charge
215       rcl _Qz_tune *                  ; rf tuning point
216       sto rfvolts                     ; save rf voltage
217
218       rcl scaled_rf
219       rcl _AMU_Mass_per_Charge *      ; multiply by mass per unit charge
220       rcl _Az_tune *                  ; substitute dc tune point
221       2 / chs                         ; additional dc factor
222       sto dcvolts                     ; save dc voltage
223
224       rcl Phaze_Angle_Deg
225       >rad                            ; degrees to radians
226       sto theta                       ; phasc angle
227
228       rcl Freqency_Hz                 ; rf frequancy in hz
229       6.28318E-6 *                    ; to radians / microsecond
230       sto omega                       ; save frequency in radians / usec
231       rtn                             ; return from subroutine
232
233
234
235   ;-------------------------------------------------------
236   seg tstep_adjust                    ; keep time step <= 0.1 Usec
237       rcl ion_time_step 0.1
238       x>y exit
239       sto ion_time_step
240
241
242   ;-----------------------------------------------------------------------
243   seg accel_adjust                    ; adds viscous effects to ion motions
244
245       rcl ion_time_step x=0 exit      ; exit if zero time step
```

```
246       rcl _linear_damping x=0 exit    ; exit if damping set to zero
247       abs sto damping                 ; force damping term to be positive
248       * sto tterm                     ; compute and save number of time constants
249       chs e^x 1 x><y -                ; (1 - e^(-(t * damping)))
250   ; factor = (1 - e^(-(t * damping)))/(t * damping)
251       rcl tterm / sto factor
252
253       rcl ion_ax_mm                   ; recall ax acceleration
254       rcl ion_vx_mm                   ; recall vx velocity
255       rcl damping * -                 ; multiply times damping and sub from ax
256       rcl factor *                    ; multiply times factor
257       sto ion_ax_mm                   ; store as new ax acceleration
258
259       rcl ion_ay_mm                   ; recall ay acceleration
260       rcl ion_vy_mm                   ; recall vy velocity
261       rcl damping * -                 ; multiply times damping and sub from ay
262       rcl factor *                    ; multiply times factor
263       sto ion_ay_mm                   ; store as new ay acceleration
264
265       rcl ion_az_mm                   ; recall az acceleration
266       rcl ion_vz_mm                   ; recall vz velocity
267       rcl damping * -                 ; multiply times damping and sub from az
268       rcl factor *                    ; multiply times factor
269       sto ion_az_mm                   ; store as new az acceleration
270
271
272
273
274   ;-------------------------------------------------------------------------
275   seg Other_Actions                   ; used to control pe surface updates
276       rcl Next_PE_Update              ; recall time for next pe surface update
277       rcl ion_time_of_flight          ; recall ion's time of flight
278       x<y exit                        ; exit if tof less than next pe update
279       rcl PE_Update_each_usec         ; recall pe update increment
280       + sto next_pe_update            ; add to tof and store as next pe update
281       1 sto Update_PE_Surface         ; request a pe surface update
282
283
284
285   ;-------------------------------------------------------------------------
286   seg Terminate
287               0 sto rerun_flym        ; turn off rerun mode
288
289
290
```

## B.2 ポテンシャルデータ取り出しプログラム

```
1   ;<div class="moz-text-flowed" style="font-family: -moz-fixed">
2   ; This PRG file was automatically generated from SL source code
3   ; using the SIS Simplified SIMION Compiler (SL) 1.0.1-2004-01-12.
4   ; WARNING: This file will be overwritten if you recompile.
5
6   ; #####
7   ; # SL program to record field data at all points on a
8   ; # 2D or 3D grid.
9   ; #####
10  ;
11  ; # The extents and granularity of the grid.
12  ; adjustable ion_x_min  = 0.5
13  DEFA ion_x_min 1
14  ;
15  ; adjustable ion_x_max  = 54.5
16  DEFA ion_x_max 28
17  ;
18  ; adjustable ion_x_step = 2
19  DEFA ion_x_step 1
20  ;
21  ; adjustable ion_y_min  = -98.5
22  DEFA ion_y_min 1
23  ;
24  ; adjustable ion_y_max  = 98.5
25  DEFA ion_y_max 28
26  ;
27  ; adjustable ion_y_step = 2
28  DEFA ion_y_step 1
29  ;
30  ; adjustable ion_z_min  = 0
31  DEFA ion_z_min 14
32  ;
33  ; adjustable ion_z_max  = 0
34  DEFA ion_z_max 14
35  ;
36  ; adjustable ion_z_step = 5
37  DEFA ion_z_step 2
38  ;
39  ; static ion_initialized = 0
40  DEFS ion_initialized 0
41  ;
42  ; static ion_jumped = 0
43  DEFS ion_jumped 0
44  ;
45  ; sub other_actions
46  SEG other_actions
47      ; ion_splat = 0  # prevent ion splats, even on electrodes
48      0 STO ion_splat
```

```
RLUP
;
; # initialize trace ion
; if ion_initialized == 0
0
RCL ion_initialized
X!=Y
GTO label19__
RLUP
RLUP
    ; ion_px_mm = ion_x_min
    RCL ion_x_min
    STO ion_px_mm
    RLUP
    ; ion_py_mm = ion_y_min
    RCL ion_y_min
    STO ion_py_mm
    RLUP
    ; ion_pz_mm = ion_z_min
    RCL ion_z_min
    STO ion_pz_mm
    RLUP
    ;
    ; ion_charge = 0  # make unaffected by fields
    0 STO ion_charge
    RLUP
    ; ion_vx_mm = 0
    0 STO ion_vx_mm
    RLUP
    ; ion_vy_mm = 0
    0 STO ion_vy_mm
    RLUP
    ; ion_vz_mm = 0
    0 STO ion_vz_mm
    RLUP
    ;
    ; ion_initialized = 1
    1 STO ion_initialized
    RLUP
    ; ion_jumped = 1
    1 STO ion_jumped
    RLUP
    ; exit
    EXIT
; begin else
GTO label20__
LBL label19__
RLUP
RLUP
LBL label20__
; end if
;
```

```
;
; # ion was jumped
; if ion_jumped == 1
1
RCL ion_jumped
X!=Y
GTO label23__
RLUP
RLUP
    ; mark()
    MARK
    ; ion_jumped = 0
    0 STO ion_jumped
    RLUP
    ; exit
    EXIT
; begin else
GTO label24__
LBL label23__
RLUP
RLUP
LBL label24__
; end if
;
;
; # jump ion to next position
; if ion_px_mm + ion_x_step < ion_x_max
RCL ion_x_max
RCL ion_px_mm
RCL ion_x_step
+
X>=Y
GTO label35__
RLUP
RLUP
    ; ion_px_mm = ion_px_mm + ion_x_step
    RCL ion_px_mm
    RCL ion_x_step
    +
    STO ion_px_mm
    RLUP
; begin else
GTO label36__
LBL label35__
RLUP
RLUP
    ; ion_px_mm = ion_x_min
    RCL ion_x_min
    STO ion_px_mm
    RLUP
    ; if ion_py_mm + ion_y_step < ion_y_max
    RCL ion_y_max
```

```
153         RCL ion_py_mm
154         RCL ion_y_step
155         +
156         X>=Y
157         GTO label33__
158         RLUP
159         RLUP
160             ; ion_py_mm = ion_py_mm + ion_y_step
161             RCL ion_py_mm
162             RCL ion_y_step
163             +
164             STO ion_py_mm
165             RLUP
166         ; begin else
167         GTO label34__
168         LBL label33__
169         RLUP
170         RLUP
171             ; ion_py_mm = ion_y_min
172             RCL ion_y_min
173             STO ion_py_mm
174             RLUP
175             ; if ion_pz_mm + ion_x_step < ion_z_max
176             RCL ion_z_max
177             RCL ion_pz_mm
178             RCL ion_x_step
179             +
180             X>=Y
181             GTO label31__
182             RLUP
183             RLUP
184                 ; ion_pz_mm = ion_pz_mm + ion_z_step
185                 RCL ion_pz_mm
186                 RCL ion_z_step
187                 +
188                 STO ion_pz_mm
189                 RLUP
190             ; begin else
191             GTO label32__
192             LBL label31__
193             RLUP
194             RLUP
195                 ; ion_splat = 1
196                 1 STO ion_splat
197                 RLUP
198                 ; exit
199                 EXIT
200             LBL label32__
201             ; end if
202             ;
203         LBL label34__
204         ; end if
```

```
205           ;
206       LBL label36__
207       ; end if
208       ;
209       ; ion_jumped = 1
210       1 STO ion_jumped
211       RLUP
212       EXIT
213   ; end segment
214   ;</div>
```

# 付 録C　Mie散乱プログラム

```
Remove["Global`@*"]                    (*全て初期化する時に使います*)
Off[General::spell];                   (*似ているスペルでもエラー表示させないためのコマンド*)
(*PolarListPlot や LogListPlot を使うために必要なコマンド*)
<< Graphics`Graphics`
(*偏向方向:ip[平行:1 _垂直:2]*)
Mie[r_: 2, nr_: 1.47, ni_: 0, dlt_: 0.22, sang_: 0, eang_: 360,
  λ_: 0.6328, ip_: 2] := (
      x = (2*π*r)/λ;               (*散乱因子屈折率 n をいれないことに注意*)
      rtd = 180/π;                 (*radian to degree ラジアンから度数表示へ変換*)
      cm = nr + \[ImaginaryI]*ni;
      snorm = 1/(π*x^2);
      nang = IntegerPart[(eang - sang)/dlt] + 1;
      dltr = Re[(π*((eang - sang)/180))/(nang - 1)];
(*sphere 関数の開始:収束させるために必要な数のセッティング*)
      xc = x + 4.05*x^(1/3) + 2;
      nc = IntegerPart[xc];
      nci = nc + 1;
      z = cm*x;
(*スタートする数の決定*)
      nmx = IntegerPart[Max[{xc, Abs[z]}]] + 15;
      an = 0;
                    For[n = nmx, n >= nc + 1, n--,
                    {
                    an = n/z - 1/(an + n/z);
                    }
                    ];
      amat[nc] = an;
                    For[n = nc, n >= 2, n--,
                    {
                    amat[n - 1] = n/z - 1/(amat[n] + n/z);
                    }
                    ];
  (*ベッセル関数の計算*)
      a = (Sin[x])/x;
      by[1] = (-Cos[x])/x;
      by[2] = (by[1])/x - a;
                    For[n = 3, n <= nci, n++,
                    {
                    rn = Re[n - 2];
                    by[n] = ((2*rn + 1)*by[n - 1])/x - by[n - 2]
                    }
                    ];
```

```
43    (*スタートする数の決定*)
44          nst = nci + IntegerPart[Sqrt[101 + x]];
45                  t[3] = 0;
46                  t[2] = 10^-35;
47                                  For[i = nst - 1, i >= nci - 1, i--,
48                                  {
49                                  t[1] = ((2*i + 1)*t[2])/x - t[3];
50                                  t[3] = t[2];
51                                  t[2] = t[1];
52                                  }
53                                  ];
54          bj[nci] = t[3];
55          bj[nci - 1] = t[2];
56                                  For[i = nci - 2, i >= 1, i--,
57                                  {
58                                  bj[i] = ((2*i + 1)*bj[i + 1])/x - bj[i + 2];
59                                  }
60                                  ];
61          alpha = a/(bj[1]);
62                                  For[k = 1, k <= nci, k++,
63                                  {
64                                  hankel[k] = bj[k]*alpha + \[ImaginaryI]*by[k];
65                                  }
66                                  ];
67    (*ベッセル関数の計算の終了*)
68          bj1 = Re[hankel[1]];
69    (*散乱展開係数の計算*)
70          For[n = 1, n <= nc, n++,
71          {
72          rf = 2*n*(n + 1);
73          bjm = bj1;
74          bj1 = Re[hankel[n + 1]];
75          b = cm*amat[n] + n/x;
76          f[n] = -\[ImaginaryI]^n*
77        rf*((b*bj1 - bjm)/(b*hankel[n + 1] - hankel[n]));
78          b = (amat[n])/cm + n/x;
79          g[n] = \[ImaginaryI]^(n + 1)*
80        rf*((b*bj1 - bjm)/(b*hankel[n + 1] - hankel[n]));
81                          If[ip == 2,
82                          {
83                          f[n] = -f[n];
84                          g[n] = g[n];
85                          }
86                          ];
87          cnrm[n] = (2*n + 1)/(rf*n*(n + 1));
88          }
89          ];
90    (*散乱展開係数の計算の終了*)
91    (*sphere 関数の終了*)
92          nci = nc + 1;
93    (*以下メイン計算*)
94      For[i = 1, i <= nang, i++,
```

```
95      {
96           (*ルジャンドル関数の計算*)
97                   θ = Re[i - 1]*dltr +sang/rtd;
98                           pnmllg[1] = 0;
99                           pnmllg[2] = 1;
100                                  For[s = 3, s <= nci, s++,
101                                  {
102                                  rn = Re[s - 1];
103        pnmllg[s] =
104         N[
105          1/(rn - 1) ((2*rn - 1)*Cos[θ]*pnmllg[s - 1] - rn*pnmllg[s - 2])];
106                                  }
107                                  ];
108          (*ルジャンドル関数終了*)
109       amp = 0;
110                  For[n = 1, n <= nc, n++,
111                  {
112                  p1 = n*Cos[θ]*pnmllg[n + 1] - (n + 1)*pnmllg[n];
113                  p2 = pnmllg[n + 1];
114                  If[ip == 1, amp = amp + I^-(n+1)*(p2*f[n] + \[ImaginaryI]*p1*g[n])*cnrm[n]];
115                  If[ip == 2, amp = amp + I^(n+1)*(-p1*f[n] + \[ImaginaryI]*p2*g[n])*cnrm[n]];
116                  }
117                  ];
118      tp = (Abs[amp])^2;
119   (*θ=θ*rtd; ポーラープロットの場合はラジアン単位でやるため度数表示には戻さない*)
120   (*LogListPlot できるが横軸がデータ数となる*)
121      resulty[i] = tp;
122      resultx[i] = θ;
123      }
124     ];
125    points =
126     Table[
127      resulty[i], {i, 1,
128       nang}];
129   (*ある半径の時のそれぞれの角度における散乱強度を代入*)
130    )
131   Do[
132          {
133          Mie[2.01 - reduction*0.01];
134                  grp = Table[{Log[10, resulty[i]], resultx[i]}, {i, 1, nang}];
135                          PolarListPlot[grp,
136                                  PlotJoined -> True,        (*線をつなげる True False*)
137                                  (*表示領域を代入 Automatic*)
138                                  PlotRange -> {{-5, 8}, {-4, 4}},
139                                  ImageSize -> 680               (*72 が 1 インチ*)
140       , PlotLabel ->StyleForm[半径 : (0.6328*x)/(2*π) μm, FontSize -> 14]
141       , TextStyle ->{FontSize -> 20}
142                  ];
143          },
144    {reduction,
145     101}                                    (*描画させたい数をいれる*)
146    ];
```

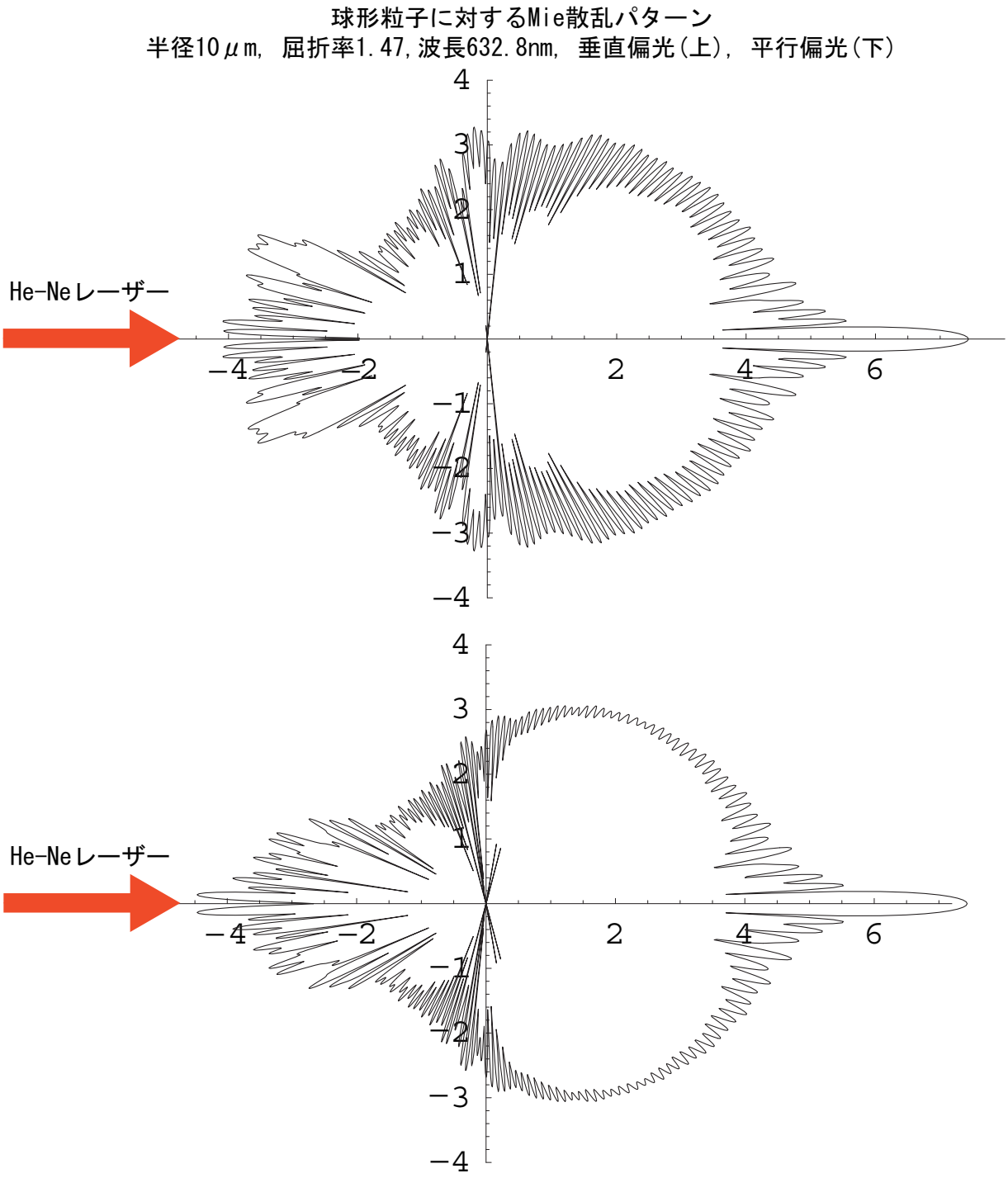



図 C.1: Mie 散乱パターンの計算値．Log スケールで書いている．前方散乱が強いのが分かる．屈折率の虚部，すなわち吸収は 0 とした．

# 付 録D　２段階励起オプトガルバノ分光

イオントラップを用いて量子計算を行うためには，レーザー冷却の技術が必要である．そしてレーザー冷却するためには分光技術が必要である．本学在籍中に分光に関してSHORT NOTESではあるが，Journal of the Physical Society of Japan **75**, 11 に掲載されたので以下に若干の解説と共に示す.

放電ガス中にレーザーを照射することで放電ガスのインピーダンスが変化することをオプトガルバニック効果という．この効果を利用したものがオプトガルバノ分光[17–19,21]であり，レーザーの波長安定化などに用いられる．波長可変半導体レーザーとHe-Neレーザーを用いて2段階励起オプトガルバノ分光をNeガスに対して行った.[20] Neガスが封入されたホローカソードランプに二つのレーザーを対向させて入射した場合と, 同方向から入射した場合とで，$^{20}$Neの同位体である$^{22}$Neの同位体シフトは，ドップラーシフトにより異なる．He-Neレーザーは$^{20}$Neの遷移 ($2p_4$-$3s_2$) で発振しているので，静止している$^{22}$Neとは共鳴しない．$\Delta\nu_{\mathrm{II}}$ だけドップラーシフトした$^{22}$Neのみが選択的に励起される．対向入射させた時，$^{22}$NeはLDを本来の周波数よりも低く感じるために高周波数側で2段励起する (Blue Shift)．逆に同方向から入射させた時は, $^{22}$NeはLDを本来の周波数よりも高く感じるために低周波数側で2段階励起する (Red Shift)．対向入射と同方向入射で共に $\Delta\nu_{\mathrm{II}}$ だけずれるので，これらの平均を取って真の同位体シフトを求めることができる．

## Apparent Isotope Shifts Observed by Two-Step Optogalvanic Spectroscopy

KEYWORD: isotope shift, neon, optogalvanic effect, Doppler shift, two-step excitation, atomic spectra

Optogalvanic spectroscopy (OGS) is a method for efficiently detecting atomic and molecular spectra. This technique has been used in frequency locking and stabilization of lasers. So far, various state-selective spectroscopies have been reported.[22–24] In this note, we suggest that the isotope shifts observed in two-step OGS are affected by the incident direction of exciting lasers, and we estimate the isotope shift of the $1s_2$-$2p_4$ transition of neon.

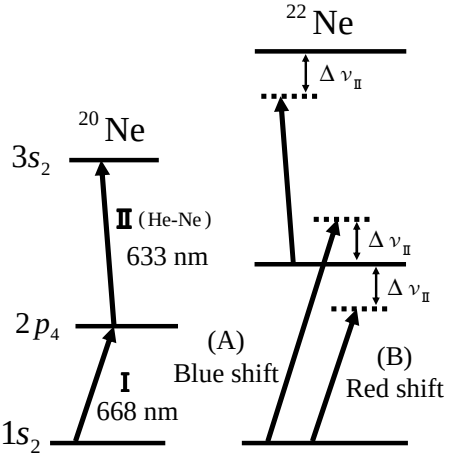



Fig. D.1: Energy levels of Ne relevant to the present observations.

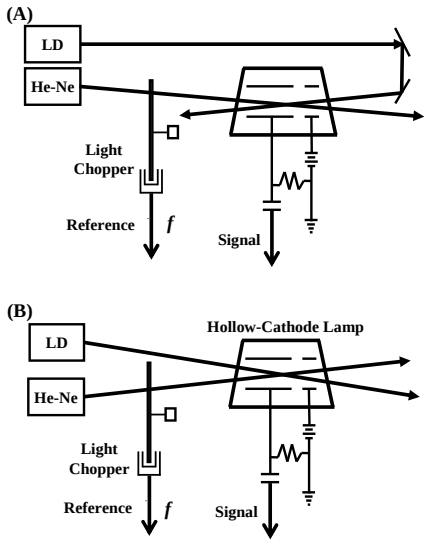



Fig. D.2: Schematic diagram of the experimental apparatus for two-step optogalvanic spectroscopy (OGS), (A) counterpropagation and (B) copropagation.

The schematic energy diagram of a neon atom is shown in Fig. D.1 We used a tunable single mode laser diode (LD) (Newport, 2010M) with a spectral width of 4 MHz, resonant with $1s_2$-$2p_4$ transition, and a multimode (consisting of 3 to 5 lines) He-Ne laser (Melles Griot, LHP925) with a longitudinal mode spacing of 257 MHz, resonant with $2p_4$-$3s_2$ transition. As shown in Fig. D.2, the two laser beams were crossed at an angle of less than 10° and were counterpropagated (case A) and copropagated (case B). A commercial hollow cathode lamp (Hamamatsu Photonics, L2783-3NE-LI) with a Ne pressure of 800 Pa was dc discharged. The discharge voltage and laser power were set to values that prevented self-absorption and self-reversal. The He-Ne laser was chopped at a frequency of $f$ (typically 330 Hz), and the two-step excitation signal was detected by a lock-in amplifier.

Figure D.3 shows the OGS signals obtained without a He-Ne laser (we call it conventional OGS in this note) and detected by scanning the LD frequency through the $1s_2$-$2p_4$ transition of Ne, along with two-step OGS signals in cases (A) and (B). The conventional OGS signal ($1s_2$-$2p_4$) is relevant to the stationary state of Ne, and the line is Doppler broadened. The isotopic component of $^{22}$Ne appears at the right foot of that of $^{20}$Ne. The resolution in two-step OGS is higher than that in conventional OGS. The central frequency of the 633 nm line ($2p_4$-$3s_2$ transition) of $^{22}$Ne is higher than that of $^{20}$Ne, and this shift is denoted by $\Delta\nu_{II}$. This value is reported to be 0.9

GHz by Stahlberg *et al.*[25] by considering that the He-Ne laser line consists of the lines only from the isotope $^{20}$Ne.

In both cases (A) and (B) in Fig. D.2, the velocity components of $^{22}$Ne atoms are toward the He-Ne laser in order to be Doppler shifted by $\Delta\nu_{II}$ and resonant with the $2p_4$-$3s_2$ transition of $^{22}$Ne atoms. In case (A), such $^{22}$Ne atoms are away from the LD

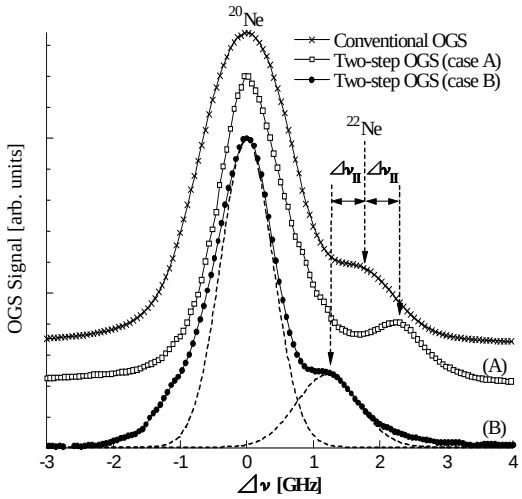



Fig. D.3: Results obtained for the $1s_2$-$2p_4$ and $1s_2$-$2p_4$-$3s_2$ transitions of Ne. Upper curve: conventional (single step) OGS. Middle (A) and lower (B), curves: two-step OGS. For case (B), the curve is deconvoluted by two Gaussian peaks.

and the frequency of the LD should be higher (blue shifted) than the central frequency of the $1s_2$-$2p_4$ transition by $\Delta\nu_{II}$. On the other hand, case (B), the two laser beams are in the same direction and the frequency of the LD should be red shifted by $\Delta\nu_{II}$. The higher resolution seen in the two-step excitation is explained by such velocity selection.

The apparent isotope shift is determined by deconvoluting the observed curve with two Gaussian peaks, as shown for case (B) in Fig. 3. The isotope shift for the $1s_2$-$2p_4$ transition should be the mean value of the two apparent shifts obtained by the two schemes of two-step OGS. The isotope shift thus obtained is estimated at $1.8 \pm 0.1$ GHz. If a He-Ne laser with a single mode is used, the peaks of two-step OGS will become much sharper and a more accurate value of the isotope shift will be obtained.

# 謝辞

本研究の進行において， 木村 正廣 教授 には多くの御指導と助言を頂き又，毎日たくさんの議論ができ，モチベーションを常に高く保つことができました．研究に行き詰まったときには，解決策のアイデアを幾度なく出して頂き，方向性がずれることなく実験を行うことができました．忙しい中，毎日多くの時間を割いて頂いた事，心から深く御礼申し上げます．

副指導教官を担当して頂いた 野中 弘二 教授 にはLDを提供して頂いたりと本研究とは直接関係ない部分でもお世話になりました．心から感謝しております．又，副審査官を務めてくださった 八田 章光 教授 にはこの場を借りて感謝の意を表します．

研究室の宮嶋泰司氏にはSIMION7.0の使い方を教えて頂きました．忙しい中，私が分からない事を調べて頂いたりもしました．シミュレーションでは大変お世話になりました．

研究室の備品管理からカメラの使用方法まで多岐に渡りお世話になった伊藤基巳紀氏に大変感謝しております．松村尚哉氏には電源装置の使い方，実験準備等お世話になりました．

郵便物を届けてもらったり，事務に関して多くの面倒を見てくれた秘書の安岡文子さん, 中山愛さん ありがとうございました．

最後に勉学に専念できるように金銭面でバックアップしてくれた家族に感謝いたします．

# 参考文献


[1] R. E. March and J. F. J. Todd: Quadrupole Ion Trap Mass Spectrometry, (Wiley-Interscience, 2005).

[2] 大槻義彦 編:「現代物理最前線３」 (共立出版社, 2000).

[3] H. J. Metcalf and P. V. D. Straten: Laser Cooling and Trapping, (Springer, 1999).

[4] M. A. Nielsen and I. L. Chuang: Quantum Computaiton and Quantum Information, (Cambridge, 2000).

[5] J. I. Cirac and P. Zoller,: Phys. Rev. Lett., **74**, 20 (1995) 4091.

[6] W.Paul, Rev. Mod. Phys. **62** (1990) 531.

[7] M. D. Barrett, et al.: Nature 429, 737-739(2004).

[8] J. Chiaverini, et al.: Nature 432, 602(2004).

[9] 松村尚哉: 高知工科大学卒業論文 (2005).

[10] 正木俊覚: 高知工科大学卒業論文 (2005).

[11] 小野田, 松村, 正木, 木村:「Mie 散乱によるマイクロ液滴の非接触精密サイズ測定」, (2005 年度応用物理学会中四国支部).

[12] 小野田, 木村：「Animation of Mie Scattering」(2006 年 Japan Mathematica Conference).

[13] C. F. Bohren and D. R. Huffman: Absorption and Scattering of Light by Small Particles, (Wiley-Interscience, 1998).

[14] P. W. Barber and S. C. Hill: Light Scattering by Particles: Computational Methods, (World Scientific Pub, 1990).

[15] H. C. Hulst: Light Scattering by Small Particles, (Dover Pubns , 1982).

[16] 戸名正英: 高知工科大学博士論文 (2002).

[17] 田内学: 高知工科大学学士論文 (2005).

[18] 鈴木淳志: 高知工科大学学士論文 (2000).

[19] 日野聡一: 高知工科大学学士論文 (2000).

[20] 小野田, 木村: 「2 段階励起オプトガルバノ分光において観測される見かけの同位体シフト」, (2006 秋季日本物理学会 25aRB-12).

[21] V. N. Ochkin, N. G. Preobrazhensky and N. Y. Shaparev: Optogalvanic Effect in Ionized Gas, (Fisec,1999)

[22] K. Miyazaki, H. Scheingraber and C. R. Vidal: Phys. Rev. A **28** (1983) 2229.

[23] J. E. M. Goldsmith: Opt. Lett. **10** (1985) 116.

[24] C. R. Vidal: Opt. Lett. **5** (1980) 158.

[25] B. Stahlberg, P. Junger and T. Fellman: Appl. Spectrosc. **43** (1989) 654.

- 本文では引用していないが，筆者が参考にした
  量子情報・コンピュータ関連の書籍

[26] H. C. von Baeyer, 水谷淳訳: 「量子が変える情報の宇宙」 (日経 BP, 2006).

[27] G. Johnson, 水谷淳訳:「量子コンピュータとは何か」 (早川書房, 2004).

[28] 石井茂:「量子コンピュータへの誘い」 (日経 BP, 2004).

[29] 古澤明:「量子光学と量子情報科学」 (数理工学社, 2005).

[30] 「量子の新世紀」 (サイエンス社, 2006).

[31] 「量子力学の発展」 (サイエンス社, 2001).

[32] 「量子情報と量子コンピュータ」 (サイエンス社, 2001).

[33] 細谷暁夫: 「量子コンピュータの基礎」 (サイエンス社, 1999).

[34] 竹内繁樹: 「量子コンピュータ」 (講談社, 2005).

[35] 根本香絵, 池谷瑠絵: 「ようこそ量子 量子コンピュータはなぜ注目されているのか」(丸善, 2006).